UNIVERSITÄT DUISBURG-ESSEN
FAKULTÄT FÜR INGENIEURWISSENSCHAFTEN
ABTEILUNG INFORMATIK UND ANGEWANDTE KOGNITIONSWISSENSCHAFT

Doktorarbeit

Development of Visualization Tools for Dynamic Networks and Evaluation of Visual Stability Characteristics

Alfredo Ramos Lezama
Matrikelnummer: DS0227273202

Abteilung Informatik und angewandte Kognitionswissenschaft
Fakultät für Ingenieurwissenschaften
Universität Duisburg-Essen

November 14, 2016

Betreuer:
Prof. Dr. Heinz Ulrich Hoppe
Prof. Dr. Jürgen Ziegler

Acknowledgements

I would like to use this opportunity to express my gratitude and appreciation to all the people who supported me during the time of my doctoral studies. To Sven Manske, Maria Frank, Fabian Bürger, and Laura Steinert, thanks for showing me that true friends always stand by your side. To Irene-Angelica Chounta, thanks for every single advice on research you gave me, all of them have been priceless. To my cousin Ada Lezama, thanks for giving me hope in all the hard moments that I faced. To my parents Joaquin Ramos and Leticia Lezama. You might have been on the other side of the world while I was doing my Ph.D. However, thanks to the strength you gave me I was able to fulfill my dreams. Lastly, I would like to express my special appreciation and thanks to my advisor Professor Dr. Heinz Ulrich Hoppe. You have been a tremendous mentor, always encouraging me to keep going and to grow from every situation. It is because of your teachings that I am finally a researcher.

Alfredo Ramos Lezama

Abstract

Dynamic graph drawings are the metaphor of choice when it comes to the analysis and visualization of dynamic networks. These drawings are often created by capturing a successive sequence of states or "snapshots" from the network under study. Then, for each one of them, a graph drawing is independently computed with the layout algorithm of preference and the resulting sequence is presented to the user in a predefined order.

Despite the simplicity of the method, dynamic graph drawings created with the previous strategy possess some problems. Actors, relations or patterns can change their position on the canvas as the dynamic network is explored. Furthermore, dynamic graph drawings tend to constantly add and remove elements without prior information. As a consequence, it is very difficult to observe how the members of the network evolve over time.

The scientific community has developed a series of layout adjustment techniques, which aim at minimizing the changes in a dynamic graph drawing. Some of them suggest that the "shape" of the drawing must be maintained at all time. Others that every actor and relation must be assigned to a fixed position in the Euclidean Space. However, a recently developed technique proposes an alternative. Multiple actors can occupy the same node position in the Euclidean Space, as long as they do not appear at the same point in time. Likewise, multiple relations can occupy the same edge position in the Euclidean Space following the principle aforementioned. As the result, a dynamic graph drawing minimizes its changes to a point where it can be perceived as *visually stable*.

This thesis presents how the visual stability of a dynamic graph drawing affects the user experience and the efficiency of the visual search when tracking actors or network attributes over time. For this purpose, a framework to support flexible visualization techniques was developed. It served as the platform to evaluate existing layout adjustment techniques. Such an evaluation combined the use of questionnaires to gather information about the user experience; an eye-tracking device to record the eye movements and a new mathematical model to appropriately quantify the visual stability of dynamic graph drawings.

The results obtained suggest that there is a trade-off between the user experience and the efficiency of the visual search, which depends on the visual stability of a dynamic graph drawing. On the one hand, dynamic graph drawings with higher levels of visual stability provide a satisfying user experience in tracking tasks. Nonetheless, they are inefficient in terms of the visual search. On the other hand, dynamic graph drawings with lower levels of visual stability, do not provide a satisfying user experience in tracking tasks but considerably improve the efficiency of the visual search. These findings were used to develop visually stable metaphors, aiming at exploring network attributes over time. Such metaphors rely on features like scaling or highlighting to improve the efficiency of the visual search.

Contents

List of Figures

List of Tables

1. Introduction

1.1. Motivation

Jacob L. Moreno was the first researcher who applied graph drawings to visualize social relations in the form of sociograms [89]. In his approach, the actors were represented as circles and the relationships between them as straight lines. Formerly, these sociograms were created using software applications known as graph editors [27]. The editors displayed the actors along with their relations in a two-dimensional Euclidean Space, in which it was possible to arrange their position manually. However, as more actors and relations appeared in a network, more effort was required to create the graph drawing [112].

Graph drawing algorithms were developed to create such visual representations without further user interaction. Moreover, they were based on a very specific set of criteria and aimed at improving the "readability" of a graph drawing, i.e., *"if its meaning is easily captured by the way it is drawn"* [112]. Thus, a graph drawing algorithm must:

- Distribute the nodes evenly.
- Minimize the number of edge crossings.
- Make the length of the edges uniform.
- Reflect inherent symmetry.
- Avoid unnecessary use of space.

The points aforementioned are known as the aesthetic criteria of a graph drawing [111]. Among the developed algorithms, the force-directed layouts cover most of the aspects established by the aesthetic criteria.

Force-directed layouts are based on physical models, where an attraction and a repulsion force interact with each other to arrange automatically the graph drawing. Eades [37] was one of the first researchers to develop such an algorithm and was inspired by a mechanical system. The algorithm considered the nodes as a set of "metallic rings" and the edges as a set of "springs". The nodes are initially distributed over the Euclidean Space by stretching the edges and then they are released. The resulting force pulls the connected nodes to each other, while the disconnected elements maintain their position. Force-directed layouts have been integrated into several software applications and, in some of them, it is possible to manipulate the graph drawing interactively [11, 12, 49, 105, 107, 114].

The interactive manipulation of a graph drawing introduces new challenges. Nodes and edges can be added or removed from the drawing after certain user interaction [112]. Their addition might produce overlaps with existing elements, while their removal

might produce a graph drawing which is completely different from the one originally displayed.

According to Eades [38], it is possible to incorporate into the force-directed layouts mechanisms to automatically adjust a graph drawing. These adjustments must maintain as much as possible the original "shape" in order to preserve the user's mental map. The mental map refers to *"the structural cognitive information a user creates internally by observing the layout of a graph"* [32]. Hence, Eades [38] developed a series of mathematical models to represent the mental map, which can be utilized with any graph drawing algorithm to adjust the drawing.

- *Orthogonal Ordering* - preserves the positions up, down, left and right of the nodes in the Euclidean Space.
- *Clusters* - preserves the position of a group of nodes in the Euclidean Space.
- *Topology* - preserves the dual graph of the original graph drawing at any time.

Eades [38] used the orthogonal ordering in the development of the force-scan algorithm. It minimizes the overlaps between the nodes and the crossings between the edges. Misue [86] applied the force-scan algorithm in several interactive graph drawings and proved mathematically that it preserved the orthogonal ordering of the mental map.

Other adjustment techniques introduced the use of constraints to achieve "stability" in a graph drawing. According to Böhringer [18], the visual representations created with a graph drawing algorithm often miss the needs of the users. Furthermore, he believed that there might be situations in which is more relevant for the user to observe nodes or edges with similar attributes on a specific location.

Force-directed layouts constantly move the nodes and edges on the canvas until a balanced drawing is created, which is expected to be aesthetically pleasing. Böhringer [18] developed a system to restrict such movements. The system was based on the use of constraints and supported three different types. *Absolute positioning* allocates the nodes a position based on a fixed coordinate system; *relative positioning* allocates the nodes a position based on the coordinates of other nodes, and *clusters* allocate a group of nodes a common location in the Euclidean Space. As more constraints are introduced into the system, the force-directed layouts decrease their movement. Böhringer [18] stated that a graph drawing whose structure does not change much when it is created for the first time can be considered as "stable". Furthermore, if the drawing satisfies the user specified constraints it is considered to have *structural stability* [18].

In some software applications, the force-directed layouts were used to display minimized overviews of larger graphs. The interaction with these visual representations often added or removed nodes, edges or even introduced successive sequences of graphs which had different layouts. The differences with the original drawing were so drastic that it was common for the users to lose their orientation on the overview [112]. Böhringer [18] introduced additional constraints based on the Euclidean Distance or the Manhattan Distance [85] aiming to achieve stability in such a scenario. However, the task was more complicated than expected because it was an unexplored research area. Since then, *"dynamic stability"* is concerned with minimizing the differences between successive layouts of graphs [18]. Eades [38], Misue [86] and Böhringer [18] are considered pioneers

in layout adjustment techniques and their findings have been used in the exploration of dynamic social networks.

Dynamic social networks are graph-based models used to study actors and their relations at different points in time. These networks can be created using two distinct representations of time. On the one hand, there is the continuous representation of time. On the other hand, there is the discrete representation of time. The discrete representation is the most applied among the researchers in the analysis of dynamic social networks. The *time window* defines the period of time in which the network will be under observation [60]. Then, the time window is used to capture a successive sequence of states or "snapshots" from the network timeline and the resulting sequence can be analyzed with social network analysis techniques.

The visual representations of dynamic social networks are known as dynamic graph drawings. These drawings are often displayed as a successive sequence of graphs due to the constant use of the discrete representation of time in the analysis phase. The The *render window* acts as a direct reflection of the time window [14]. Thus, for each of the captured snapshots, a graph drawing is computed separately with the algorithm of choice and the resulting sequence is presented to the user in a predefined order.

The visual attributes of the nodes and edges in a dynamic graph drawing can be adjusted to identify patterns in a dynamic social network. For instance, nodes can increase or decrease their radius based on numeric attributes or even use different color schemes to match non-numeric attributes [19–21]. The edges can increase their thickness, use different color schemes or even distinct transparency levels [1, 6, 69].

In spite of the visual enhancements, the dynamic graph drawings still present some issues. The algorithms compute a different graph drawing for each snapshot in the sequence. Hence, it is very likely for actors, relations, or patterns to change their position on the canvas during the exploration of the dynamic social network. Moreover, the dynamic graph drawings are constantly adding or removing nodes and edges at different points in time. The combination of these factors makes it very difficult to track something over time and can disturb the user to the point of breaking the mental map.

Modern software applications to visualize dynamic social networks combine graph drawing algorithms with layout adjustment techniques. These techniques aim at minimizing the chances in a successive sequence of graph drawings, either by *maintaining a constant shape* [7], *decreasing the movement of the nodes and edges* [40] or *reducing their constant addition and removal* from the canvas [31]. Layout adjustment techniques have been using characteristics mentioned in the theories of visual stability, which are related to the origin of the mental map.

According to some of them [25, 82, 83], on a given image, each saccade attempts to direct the eye-gaze towards specific elements which are scanned in detail. At the same time, a mental representation of the location with the scanned elements is generated. In other words, the mental map. As new visual information becomes available, the next saccades produce a search on the region where the scanned objects were located. Therefore, the sense of a stable image occurs if it is possible to locate the scanned elements. The principles mentioned in the theories of visual stability can be used to develop layout adjustment techniques that allow the users to effectively and efficiently track actors or other patterns in a dynamic social network.

1.2. Scientific context: the SISOB project

People around the world are becoming more connected between each other due to the use of platforms like social networks, social media, and other online resources. The information available on these platforms is able to travel around the world in a matter of minutes, affecting the society itself in many different ways. Positive articles, news or notes have a positive impact on people. Negative newsletters, articles or posts tend to have the opposite effect. In a world where the information has such influence, could it be possible to measure or quantify appropriately the impact of science in society?. The European Union project SISOB aimed to provide an answer to this question, developing a technological observatory for science based on social models [1].

Decision makers, along with funding agencies, try to support researchers whose work can have a high impact on society. The evaluation of viable funding candidates involves the results of the peer reviewing process before an article has been published and the values of the scientometric indicators once the article has been published. However, this procedure only evaluates the quality of the work in the context of a scientific community rather than the impact it has on society. The SISOB project provided software solutions to quantify the social appropriation of the scientific knowledge. The solutions are based on social network analysis techniques which were used in three different case studies.

Researcher mobility: The researchers frequently change their location to collaborate with other researchers around the globe. As the researchers move, it is possible for individuals, laboratories or institutions to appropriate knowledge which might not have been available before. The mobility of a researcher has allowed the creation of networks that contain information about culture, methods, and techniques of several researchers. The SISOB project used graph models to represent and analyze such networks. The models took into account the visits to laboratories, lectures, and long research exchanges.

Knowledge sharing: The social appropriation of knowledge completely depends on the will of a researcher to share his knowledge with other persons. For example, colleagues, scientists working on a distinct discipline or even the public in general. The process of sharing scientific knowledge is not restricted only to the scientific publications and the conference presentations. It is possible to use other communication channels to reach the public. Blogs, newspaper or interviews broadcasted through radio and television can serve as a bridge between the people and the researchers. Furthermore, the existence of these communication channels offers the opportunity to model networks of actors and scientific artifacts.

Peer reviewing: The peer reviewing mechanisms for scientific publications, conference presentations, and funding proposals were originally designed to guarantee high-quality scientific articles. Nevertheless, the same mechanisms have been criticized for failing in the identification of works with high potential impact. The risk of failing in the identification process increases if the scientific discipline is dominated by a group of prestigious reviewers that only evaluate the work of each other and reject novel ideas. From the perspective of the SISOB project, the peer reviewing process can be seen as a reviewer-author network and can be further analyzed to improve the reviewing process.

[1] http://sisob.lcc.uma.es/

The SISOB project had two major outcomes in terms of software solutions to conduct the case studies. On the one hand, there is the web-based analytics workbench [55]. On the other hand, there is the framework to support flexible visualization techniques. The web-based analytics workbench is a software application designed to create and execute network analysis workflows. It combines a web-based user interface inspired by the pipes-and-filters metaphor with a server-side computational backend. The backend relies on an open source implementation of the tuple space concept [50] and a multi-agent system to perform the actual analysis. The framework to support flexible visualization techniques is designed to complement the analysis functions of the workbench. It counts with a web-based user interface that allows the interactive manipulation of dynamic graph drawings. At the same time, its drawing engine uses exclusively scalable vector graphics and can make use of the multi-agent system of the workbench to compute either partially or completely the visual metaphor. The framework did not only serve as a platform to visualize the results of the case studies but also to study the implications of visual stability when visualizing dynamic social networks.

1.3. Definition of the problem

Minimizing the changes in a successive sequence of graph drawings has become one of the highest priorities in the visualization of dynamic social networks. Mainly, because according to some studies, achieving "stability" on the visual level improves the identification of single elements, [99], the memorability of the network [4] along with the user orientation on the canvas [5]. Yet, there is a factor that has never been considered before as part of an evaluation.

Does the visual stability of a dynamic graph drawing improves the efficiency of the visual search and the user experience when tracking actors or other patterns in a dynamic social network?.

This thesis aims to determine if the visual stability of a dynamic graph drawing helps the users to effectively and efficiently track actors or other network related patterns over time. In comparison to other empirical investigations, this thesis emphasizes the efficiency of the visual search along with the user experience, using the theories of visual stability as the basis to find an answer to the question described beforehand. In order to proceed in this direction, it is necessary to take into account some technical and algorithmic limitations.

Nowadays many software applications to visualize dynamic social networks are available. Some of them are designed to be executed on a local machine, while others rely on a web server for their execution. The software applications that run on a local machine offer graph drawing algorithms [52], layout adjustment techniques [7], and features to enhance the dynamic graph drawing [10]. In addition, they support distinct formats to encode network information. The software applications that run on a web server have different characteristics. Usually, they are mounted on top of large databases and can extract from there different types of networks [62]. Furthermore, the dynamic social networks are illustrated with more complex metaphors rather than traditional graph drawings [47].

Both classes of software applications present useful features to visualize dynamic social networks, but also have their own technical limitations. A first technical limitation appears when visualizing large dynamic social networks. The drawing engine is the component responsible for creating the visual representation of the network under study [10] and its performance depends entirely on the resources of the computer. In the case of the software applications that run on a local machine, the performance of the drawing engine decreases as the data becomes larger. The software applications that run on a web server have more resources at their disposal. Nonetheless, that does not necessarily solve the problem. The web technologies to create visualization techniques always compute the drawing on the web browser and only in rare occasions the drawing engine has access to the resources of the server. Thus, the rendering of the drawing becomes slower as the size of the data set increases.

A second technical limitation appears when it comes to the extensibility of the software applications. Those applications designed to be executed on a local machine have well-defined interfaces to integrate new layout adjustment techniques and visual metaphors [10]. Nevertheless, the performance of the drawing engine might not be adequate to study the implications of visual stability. The software applications that run on a web server have more resources. Unfortunately, they do not provide a set of well-defined interfaces to integrate new approaches nor instructions to adjust the performance of the drawing engine [47, 62]. Therefore, it is necessary to develop a flexible visualization framework addressing the two technical limitations.

The algorithmic limitation concerns the metrics to quantify the changes in a dynamic graph drawing and the layout adjustment techniques. Metrics like the Euclidean Distance or the Manhattan Distance can be used over a successive sequence of graph drawings to quantify their differences. Such a strategy assumes that each actor appearing in the dynamic social network is mapped to a unique position in the Euclidean Space. However, not all the dynamic graph drawings use the same mapping. Diehl [31] developed a dynamic graph drawing algorithm called *Foresighted Graph Layout*. The approach is able to map multiple actors to the same node position in the Euclidean Space as long as they belong to different points in time. Likewise, multiple relations are mapped to the same edge position in the Euclidean Space as long as they appear at different points in time. Additionally, the Foresighted Graph Layout can be combined with any graph drawing algorithm and layout adjustment technique. The existing metrics are not able to quantify appropriately the changes in a drawing produced with the Foresighted Graph Layout due to its mapping. Thus, a more elaborated model is needed to cover approaches like the one proposed by Diehl [31].

Case studies have been conducted to determine the advantages of adjusting the layout of a dynamic graph drawing [4, 5, 98, 99]. Still, none of them has ever considered the Foresighted Graph Layout nor the particular characteristics of the drawing it produces. During the exploration of the dynamic social network, the dynamic graph drawing has the minimal number of changes in its structure. This is because the mapping employed by the Foresighted Graph Layout prevents the nodes and edges to be constantly added or removed from the canvas. Furthermore, it assigns the nodes and edges a fixed position in the Euclidean Space. As a result, the dynamic graph drawing is perceived as a visually stable image. A case study that involves eye-tracking devices [63] and questionnaires can help to determine the implications of visual stability in a dynamic graph drawing.

1.4. Achievements

The thesis presents software and algorithmic solutions to the limitations described above. The main objective is to determine if the visual stability of a dynamic graph drawing helps the user to effectively and efficiently track actors or other network related patterns over time. In a first instance, the framework to support flexible visualization techniques developed during the SISOB project addresses the technical limitations. Its architecture allows the integration of graph drawing algorithms, layout adjustment techniques as well as other visual metaphors with ease. Furthermore, the drawing engine can use the multi-agent system of the web-base analytics workbench [55] to increase its performance depending on the complexity of the visual metaphor.

As a response to the algorithmic limitations, a mathematical model was developed using the theories of visual stability as its basis. The approach is able to accurately quantify the changes in those dynamic graph drawings created with the discrete representation of time, including the Foresighted Graph Layout. A case study was conducted to evaluate state of the art layout adjustment techniques. It combined the use of the model to quantify the visual stability of distinct dynamic graph drawings; questionnaires to gather information about the user experience and an eye-tracking device to evaluate the efficiency of the visual search. The results obtained from the case study were used to develop new layout adjustment techniques and visual metaphors to track actors, relations and other patterns in a dynamic social network. The details on these approaches are presented in the subsequent chapters of this thesis.

1.5. Structure of the thesis

This thesis is organized in nine chapters:

1. **Introduction:** Motivation, definition of the problem and the outline of the thesis.
2. **State of the Art:** State of the art regarding dynamic networks, visualization of dynamic networks, layout adjustment techniques and theories of visual stability.
3. **A framework to Support Flexible Visualization Techniques:** Concept, architecture and features of a framework to support flexible visualization techniques in a web-based analytic environment.
4. **A Mathematical Model of Visual Stability for Dynamic Graph Drawings:** Foundations and metrics that form the mathematical model of visual stability for dynamic graph drawings.
5. **Evaluation of the Mathematical Model of Visual Stability:** Case study to evaluate the mathematical model of visual stability.
6. **Algorithms to Support Visual Stability in Dynamic Graph Drawings:** Foundations and characteristics of two algorithms to support visual stability in dynamic graph drawings.

7. **Evaluation of the Algorithms to Support Visual Stability in Dynamic Graph Drawings:** Case study to evaluate the algorithms to support visual stability in dynamic graph drawings.
8. **Exploring Network Attributes in Visually Stable Representations:** Visually stable representation designed to track the productivity and collaborativeness of researchers in a dynamic scientific network.
9. **Conclusions:** Summary of the thesis, conclusions and possible future directions of this research work.

This thesis is partly based on a set of scientific peer-reviewed articles published in journals and in the proceedings of international conferences. The following articles are particularly referred to:

- *Ramos, A., Chounta, I.-A., Göhnert, T., and Hoppe, U. (2015). Visual stability in dynamic graph drawings. Journal of Interactive Media (I-COM).* - Chapter 5
- *Ramos, A., Chounta, I.-A., Göhnert, T., and Hoppe, U. (2015). Exploring visual stability in dynamic graph drawings: A case study. In proceedings of the 2015 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM).* - Chapter 4
- *Ramos, A., Chounta, I.-A., Göhnert, T., and Hoppe, U. (2015). Using visual stability to support search efficiency and user experience in dynamic graph drawings. In proceedings of the Network Intelligence Conference (ENIC).* - Chapters 6 and 7
- *Ramos, A., Göhnert, T., Zeini, S., and Hoppe, U. (2014). Foresighted heat ring: Tracking the productivity and collaborativeness of researchers in dynamic networks. In proceedings of the Network Intelligence Conference (ENIC).* - Chapter 8

2. State of the art

2.1. Introduction

This chapter presents the state of the art regarding the analysis and visualization of social networks. Both topics have evolved alongside each other, providing metrics, algorithms and different types of visual representations. However, a single network only represents the state of this structure at a specific point in time. This chapter also includes the fundamental concepts behind the analysis and visualization dynamic social networks. In addition, it presents psychological theories of visual stability and their relationship with the mental map. Lastly, layout adjustment techniques for minimizing the changes in dynamic graph drawings are discussed.

2.2. Social networks

A social network consists of a set of actors connected between each other through a set of social links. It is formally defined as a graph-based model in the form $g = (V, E)$, where V refers to the a set of vertices and E to a set of edges $E \subseteq V \times V$ [117]. Over the years, the scientific community has developed metrics, indicators, and algorithms to study the patterns that are hidden within a social network. For example, Bavelas [13] worked on an indicator to detect relevant actors. He derived the notion of relevance from how close an actor is to the other members in the social network and applied the concept to create the closeness centrality. Closeness centrality calculates the sum of all shortest paths from a given actor to all the other members in a social network. Therefore, with a shorter distance, an actor is considered to be more relevant.

Freeman [44] also worked on the detection of relevant actors. However, he derived the notion of relevance from two different concepts. The first one considers an actor to be relevant depending on the number of connections established on the social network. This was applied in the development of the degree centrality, which calculates the sum of all connections established by a given actors in a social network. With more connections, an actor is considered to be more relevant. The second one considers an actor to be relevant depending on the number of times he or she intervenes in the communication between two other actors in the social network. This was applied in the development of the betweenness centrality [45]. The centrality indicators might have been developed in a social context, but have been used to identify other patterns in a variety of networks.

Crucitti [29] studied the organization of urban areas from several cities around the world. In his approach, the urban areas were mapped into a network model. The edges of the network represented the streets and the nodes their intersections. In addition, the network model was analyzed using distinct centrality indicators like degree, closeness, betweenness, straightness and information. Straightness centrality [122] suggests that

an efficient communication between a node a and a node b is inverse to the shortest path length $d_{i,j}$. Information centrality [73] considers a node to be important depending on the performance of the network before and after the deactivation of the node. If the network performance decreases, the relevance of the node is higher [72]. The results of the study were presented in distinct categories; each one related to a centrality indicator. Nonetheless, the degree centrality was omitted because it did not provide a meaningful outcome. Closeness centrality obtained higher scores at the center of the urban areas due to a short distance between the intersections of the streets. Betweenness centrality obtained higher scores on prominent urban routes across a number of intersections, changes in direction and also focal urban spots. Straightness centrality obtained high scores on main streets and even higher values on their corresponding intersections. Information centrality had similar results to the betweenness centrality.

In addition to the relevant actors, social networks also tend to hide social groups within its structure. The notion of a social group has been used widely in social sciences, such as social psychology and sociology [117]. The researchers working in these subjects often avoid providing a more precise definition because "everybody knows what it means" [117]. Nonetheless, the network scientists who have worked in the development of algorithms to detect social groups provide a more concrete definition [2, 87, 109]. According to them, the concept of a social group or subgroup can be formalized by looking at the cohesion among its members and can be studied using the properties of the ties connecting the members. Therefore, a cohesive subgroup is defined as a subset of actors that tend to be more densely connected to each other in comparison to other actors in a social network [9, 118]. These groups are characterized by four properties [117]:

- Mutuality of ties.
- Closeness or reachability of its members.
- Frequency of ties among members.
- Relative frequency of ties among subgroup members compared to non-members.

Festiger [42], Luce and Perry [80] studied social networks to characterize cohesive subgroups. In their studies, the actors were connected to each other through a set of friendship links. It was discovered that the actors tend to find groups of three members minimally and the members happen to be mutual friends. Such a finding introduced one of the fundamental concepts to identify cohesive subgroups: "a clique". A clique is defined as a maximally connected subgraph of three or more nodes [117]. The nodes in the clique are adjacent to each other and there are no other nodes adjacent to the members in the clique. A clique is a useful starting point in the identification of a cohesive subgroup. Nonetheless, its definition is too strict. The absence of a single link can avoid a cohesive subgroup to be detected.

N-cliques [2] provide a more flexible definition since the maximality condition established by the clique is ignored. They are based on reachability and require all the members of a cohesive subgroup to have a certain geodesic distance between them. The geodesic distance is the length of the shortest path between a pair of nodes. Thus, more formally, an n-clique is a maximal subgraph in which the largest geodesic distance between any two nodes is not greater than n [117]. Other flexible definitions of the original clique are the n-clans and the n-clubs [87]. These are based on diameter rather than reachability.

An n-clan is an n-clique, but it requires the geodesic distance of all nodes in the subgraph to be lower or equal than the value of n for all paths within the subgraph [117]. An n-club is defined as a maximal subgraph of diameter n [117].

Other definitions of cohesive subgroups require the members to have a minimal number of connections. K-plexes, for instance, are maximal subgraphs containing i nodes. Each node must be adjacent to no fewer than $k - i$ nodes within the same subgraph [117]. In other words, k-plexes specify an acceptable number of connections that can be absent for each node. K-cores are more restrictive than k-plexes. They are maximal subgraphs in which a node must be adjacent to a minimum of k nodes within the same subgraph [108]. According to Dorogovtsev [34], k-cores can be calculated with a very simple algorithm. First, all the nodes with a degree lower than k must be removed from the network. As next step, it is necessary to check the degree of the remaining nodes. If the degree is lower than k, the node must be removed from the network as well. This process is repeated until it is no longer possible to remove any other node. As a result, the k-cores in the network are obtained.

Palla et al. [92] suggested that, in real world networks, cohesive subgroups might overlap through a set of common members. This assumption led to the development of the clique percolation method. The clique percolation method uses the topological properties of the network in order to detect overlapping communities. A k-clique is considered as a set of nodes k, in which every member of the set is connected to each other. Hence, a k-clique community consists of a set of nodes that can be reached through a set of overlapping k-cliques. In this context, an overlap means that the k-cliques share minimally k-1 nodes.

An implementation of the clique percolation method was proposed by Kumpula et al. [71]. It is executed in two separate phases. The first phase removes the edges from the given network and inserts them one by one in a subsequent step. Here, the objective is to detect the formation of the k-cliques when the edges are inserted. An edge always involves a connection between a node v and a node u. In order to detect a k-clique, it is necessary for the nodes v and u to have a degree of $k-1$. Then, the neighbors of v and u are used to form the neighborhood $N_{uv} = N_u \cap N_v$. Every $k-2$ clique contained in N_{uv} forms a new k-clique and these are used as the input for the second phase. The second phase creates a bipartite network using one by one the detected k-cliques. A "tag node" is inserted to denote the name of a given k-clique and all the nodes in the k-clique are connected to this "tag node". As a new k-clique is transformed into a bipartite graph, its members start to connect with multiple "tag nodes". These connections represent the overlaps between the k-cliques. In some scenarios, the networks are modeled using distinct types of nodes as in a bipartite graph.

In the analysis of social networks, a one-mode network is a graph in which all the nodes belong to one same type and these nodes are connected to each other without any restriction. A two-mode network, on the other hand, is a graph in which the nodes belong to two different types and the connections between them are established as in a bipartite network. The nodes of a type A are never connected among themselves, but only to those nodes of a type B. Likewise, the nodes of a type B are only connected to the nodes of a type A and never among themselves. Two-mode networks are useful to model cases in which the actors interact with other entities.

For example, a scientific network [41] consists of a set of researchers who have worked on common scientific artifacts. These artifacts can be the prototype of a machine, a book, a book chapter, an article, etc. A scientific network can be modeled as a two-mode network. The researchers are never connected among themselves, but only to the scientific artifacts. Similarly, the scientific artifacts are never connected to each other, but only to the researchers. The productivity and collaboration of researchers are attributes often analyzed in a scientific network [75]. The former is the number of artifacts produced by a single researcher, whereas the latter is the number of colleges a single researcher has worked to produce a scientific artifact [41]. The productivity of the researchers can be calculated by applying the degree centrality on the two-mode network. However, the collaboration requires a different procedure.

A graph folding operation is used to transform two-mode networks into one mode-networks, with the disadvantage of losing some information. It proceeds by multiplying the adjacency matrix of the original two-mode network and its transpose. Nonetheless, it is also possible to inverse the order of the matrices in the multiplication. As a result, two types of one-mode networks are obtained. One contains the scientific artifacts and the connections between them. The other one contains the researchers and their relations. The degree centrality of the researchers in such a network can be used as an indicator for the collaboration. Indifferently of the analysis or the network under study, a visual metaphor often helps to have a better understanding of the discovered pattern [43].

2.3. Visualization of networks and visual analytics

Visual analytics is a scientific discipline that emerged from information visualization and scientific visualization [123]. It promotes the analytical reasoning through the use of interactive visual interfaces in order to integrate the users in the analysis process [123]. Visual analytics is considered as a multidisciplinary field since it brings together scientific and technical communities such as computer science, information visualization, cognitive sciences, interactive design, graphic design, and even social sciences [33]. The goals of visual analytics tend to overlap with those established in information visualization and scientific visualization. However, visual analytics is mainly concerned with the application of human judgment to reach a conclusion from the observation of evidence [113]. The analysis of a social network results in data related to a pattern under study. The data often comes in the form of numeric or non-numeric attributes, which can be further analyzed and better interpreted using graph drawings.

The drawing d of a graph g is defined as a mapping of each vertex v from g to a distinct point $P(v) = (v_x, v_y)$ of a plane and each edge (u, v) from g to a simple Jordan Curve with end points $P(u)$ and $P(v)$ [111]. A straight line drawing is one in which every edge is mapped to a straight line segment, or more formally, a straight line drawing is an injective function $f : v \in V \rightarrow (v_x, v_y) \in \mathbb{R}^2$ [111]. This definition has been used in the development of graph drawing algorithms and also visualization techniques to support the analysis of social networks.

Kamada and Kawai [67] developed a graph drawing algorithm inspired by the mechanical system of Eades [37] as it is described in Chapter 1. According to them, the ideal distance

in the drawing for a pair of nodes that is connected through an edge is equal to their theoretic distance in the corresponding graph [67]. The term theoretic distance refers to the length of the shortest path between the two nodes [68]. Thus, the algorithm considers an edge as a spring with an arbitrary length. The objective is to match the length of the spring to the length of the shortest path between the two nodes. If the nodes are too far away from each other, the spring applies an attraction force. Otherwise, a repulsion force is utilized. Such a procedure is executed for all edges in the drawing. The ideal matching is often difficult to obtain and it is recommended to operate at an intermediate distance to improve the performance of the algorithm [68].

Fruchterman and Reingold [46] also developed a graph drawing algorithm inspired by the work of Eades [37]. Nonetheless, their approach considers the nodes as a set of "atomic particles" or celestial bodies [68]. Each of such particles produces two different types of forces. An attraction force is produced by those adjacent particles, whereas a repulsion force is produced by all the other particles in the system. In addition, Fruchterman and Reingold [46] introduce the notion of temperature as part of the algorithm. This parameter acts as a control mechanism to the described forces. Higher temperatures allow the forces to adjust the distance between the particles with more efficiency, but every adjustment makes the temperature to decrease. As a result, the forces perform fewer adjustments to the elements in the graph drawing which eventually stop moving.

Jacomy et al. [65] developed the graph drawing algorithm known as ForceAtlas2. It is inspired by the work of Eades [37] and it elaborates on the ideas introduced by Fruchterman and Reingold [46]. ForceAtlas2 considers the nodes as a set of charged particles. Each one of them produces an attraction force to all those particles that are connected through the same edge. At the same time, each particle produces a repulsion force to all those that do not satisfy the last condition. The intensity of the repulsion force depends on the number of connections and such a "repulsion by degree" [65] is calculated as follows:

$$FA_{fr}(v,u) = k_z \frac{(deg(v)+1) * (deg(u)+1)}{dist(v,u)}$$

where k_z is a numeric value defined by the user. According to Jacomy et al. [65], the interaction between the two forces produces a graph drawing in which is possible to observe groups of highly connected nodes. These groups also maintain an appropriate distance from each other due to the use of a *gravitational force*. It prevents the elements in the graph drawing to move far away from the center of the canvas. The gravitational force is calculated as follows:

$$FA_{fg}(v) = k_g * (deg(v)+1)$$

where k_g is a number specified by the user. In some situations, the attraction and gravitational forces can overcome the repulsion force. ForceAtlas2 allows the use of edge-related attributes to enhance this last one. The attribute acts as an additional constant in the equation for the repulsion force.

Krempel [69] developed a graph drawing algorithm taking into account the attributes of the actors in a social network. His approach, the Simple Space Solution, presents the actors as nodes and the relationships between them as edges. The nodes are distributed around a circumference in clockwise order. However, their position depends on the attribute to visualize. The attribute is used as a parameter to sort the nodes in descending order. Thus, the Simple Space Solution algorithm assigns the first positions around the circumference to those actors with higher values according to the attribute and those actors with lower values are assigned to the subsequent positions. As an example, let's assume that the attribute to visualize is the degree centrality. The Simple Space Solution algorithm will assign the first positions around the circumference to those actors with a higher degree, while the subsequent positions will be used to place the actors with a lower degree.

Krempel [69] has also developed graph visualization techniques to illustrate the attributes of the entities in a network. In his approach, the Krempel Pie Charts, the entities are illustrated as nodes and the relationships between them as edges. The nodes present special characteristics based on the attributes of the entities. On the one hand, the non-numeric attributes are used to create a color scheme for the nodes. On the other hand, the numeric attributes are used to create pie charts which are displayed inside the corresponding nodes. The edges also present special characteristics based on the attributes of the relationships. The non-numeric attributes are utilized to create a color scheme for the edges, while the numeric attributes are used to increase their thickness.

Bastian et al. [10] developed Gephi, a software application to support the analysis and exploration of large networks. Gephi has a unique rendering engine. It exclusively uses the graphics processing unit and is capable of creating graph drawings with over 20,000 nodes. In addition, the rendering engine uses the ForceAtlas2 algorithm [65] by default to distribute the nodes and edges in the Euclidean Space. Gephi also offers styling options to adjust the visual appearance of the graph drawing. The nodes can take any shape, use textures or even display a picture. The edges can change their color scheme, vary their thickness and even display labels with additional information. Figure 2.1 illustrates the software application Gephi.

The analysis and visualization of a social network capture the actors, relations, and patterns that exist at a specific point in time. In recent years, there has been a growing interest among the researchers to study how these elements evolve. Thus, the notion of a static social network was extended to a dynamic social network.

2.4. Dynamic social networks

Dynamic social networks are graph-based models used to represent actors and their relations at different points in time. Time is what makes these networks so important. It allows the researchers to observe the moment in which an actor, relation or pattern emerges, evolves and vanishes from the network. Dynamic social networks are based on three concepts. First, there is the timeline, which is the overall period of time the network information has been recorded [14]. Second, there is the time window [60]. It is an interval, whose start point and end point defines the period of time in which a

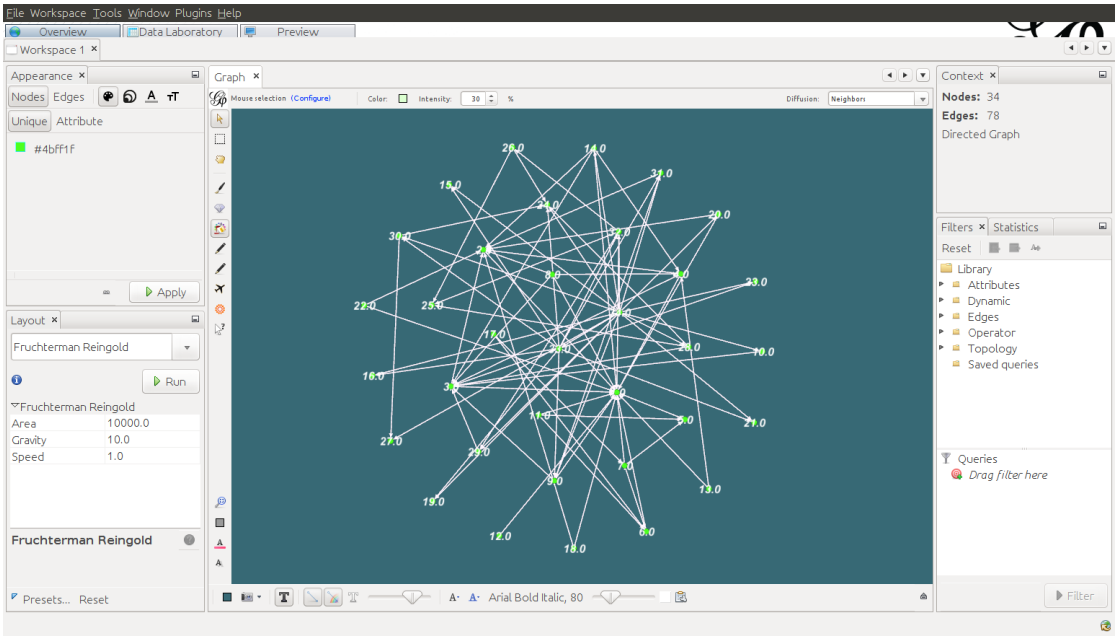

Figure 2.1.: Gephi

network will be under observation [60]. Third, there is the representation of time. Time can be represented in two distinct forms. It can be continuous or discrete. Both forms use the timeline and the time window to create a dynamic social network.

The continuous representation of time [14] considers the actors and their relations as events with an exact starting and ending time. The events appear with a specific duration on the network timeline, simulating a "stream" of information. The time window is used to capture the elements in the stream. All those actors and relations whose duration falls within the time window form the dynamic social network. The dynamic social networks created with the continuous representation of time feature a high level of detail. It is possible to observe the exact moment in which an actor or relation appeared and disappeared from the network. However, such a level of detail produces very large data sets and in some cases, they are difficult to handle. Figure 2.2 illustrates the continuous representation of time.

The discrete representation of time [14] produces more compact data sets. The actors and their relations are considered to exist only at a certain point in time rather than having a time duration. As illustrated in Figure 2.3, this approach captures a successive sequence of "states" or "snapshots" from the network timeline. All the actors and relations which have been captured in the sequence of the snapshots from the dynamic social network. Such an approach is the most used among the researchers to study dynamic social networks. Moreover, it has been used as the basis to develop indicators and algorithms to detect patterns over time.

For instance, Trier and Bobrik [115] developed an indicator to detect relevant actors in a dynamic social network. In their approach, the notion of relevance comes from how much an actor contributes to the connectivity of the network. A contribution is considered to be meaningful if the length of the paths between any pair of actors is

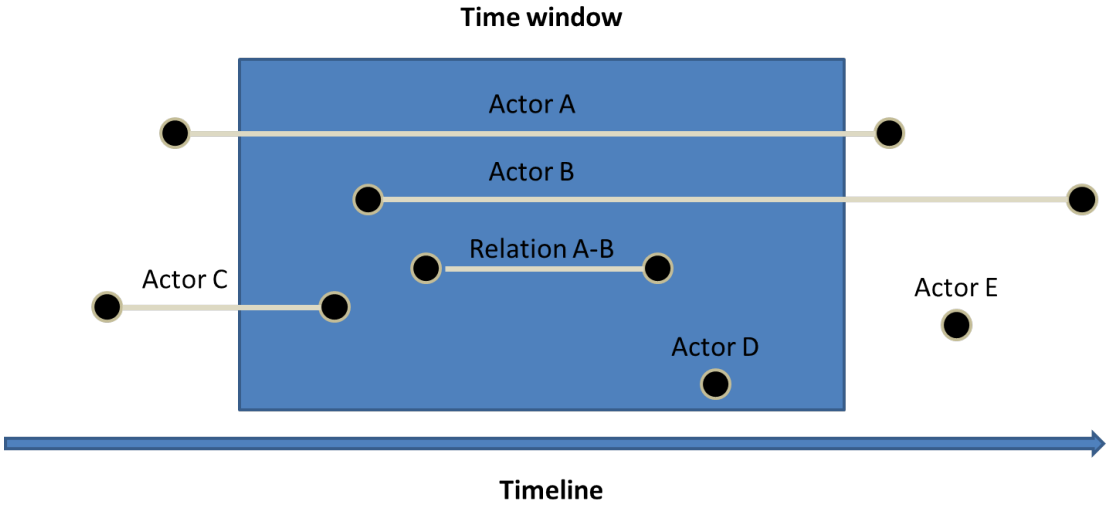

Figure 2.2.: Continuous representation of time

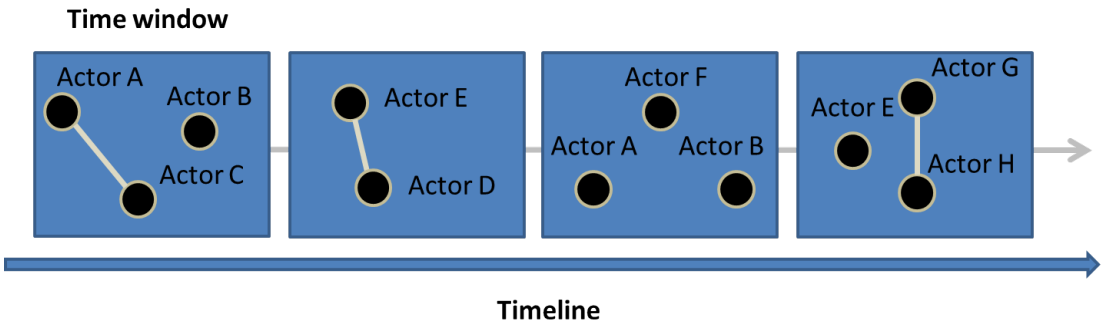

Figure 2.3.: Discrete representation of time

reduced. Therefore, the "brokering activity" indicator compares such a characteristic in the presence and absence of the desired actor at a given point in time. More formally:

$$BA(g^t) = \sum_{u \in V(g^t)} \sum_{v \in V(g^t)} f_b(p'_{u,v} > p_{u,v})$$

where u and v are a pair of actors in the dynamic social network at the same point in time, $p'_{u,v}$ is the shortest distance between u and v in the absence of the desired actor, $p_{u,v}$ is the shortest distance between u and v in the presence of the desired actor and $f_b(p'_{u,v} > p_{u,v})$ is a boolean function in the form:

$$f_b : (A, B) = \begin{cases} 1, \text{ if } A > B \\ 0, \text{ otherwise} \end{cases}$$

Trier and Bobrik conducted a study to evaluate the efficiency of their indicator [115]. It was compared against the degree centrality and the betweenness centrality. The data used in the study was the corporate e-mail network of Enron managers. The network contained 4526 e-mail messages, 112 managers from 01/04/2000 to 21/30/2000. According to the results obtained, the brokering activity indicator was able to detect actors who contributed more in the connectivity of the network. These actors were never detected by the degree centrality nor the betweenness centrality.

Horak et al. [62] worked on a series of indicators to detect relevant actors in a dynamic co-authorship networks. According to them, these networks have the capability to remember the interaction between any pair of researchers. This comes from the fact that frequent interactions stabilize the relationship between the researchers over time. The stability of a relationship (S_{rls}) refers to the number of periods of time in which a relationship between any pair of researchers is active. Likewise, the stability of a researcher (S_{rch}) refers to the number of periods of time in which a researcher is active. Both concepts are the basis of the developed indicators. Cooperativeness calculates the number of interactions established between a given researcher and its immediate neighborhood. The indicator is defined as follows:

- Let N_{rls} be the set of all the relations in which a researcher r has been involved.
- Let N_{rch} be the set of all researchers that have worked with a researcher r.

$$Cooperativeness(r) = \sqrt{\sum_{rls \in N_{rls}} S_{rls}(rls) * \sum_{rch \in N_{rch}} S_{rch}(rch)}$$

Solidity calculates the strength of the interactions between the researchers and their immediate neighborhood. It is defined as:

$$Solidity(r) = \sum_{rls \in N_{rls}} (S_{rls}(rls) - MinS_{rls})$$

where $MinS_{rls}$ is a minimum level of stability. The indicators developed by Horak have been used in Forcoa.net [62] to explore the DBLP data set from the field of computer science. The system is explained in the subsequent section of this chapter.

One of the biggest challenges in the analysis of dynamic social networks is the detection of cohesive subgroups. According to Palla et al. [91], most difficulties to be solved are due to a set of characteristics that appear in a dynamic scenario. A cohesive subgroup can emerge, grow, contract, split or even disappear without prior information. A common strategy to detect them is to use the algorithms designed for static social networks and match the resulting groups over the different periods of time.

Lehmann et al. [76] proposed an extension of the clique percolation method [92] to detect cohesive subgroups in a bipartite social network at a specific point in time. A bipartite network is defined as a non-overlapping set of nodes δ and γ, where the edges connect exclusively the elements of both sets. A $K_{a,b}$ clique is a complete subgraph with a subset of nodes a from δ and a subset of nodes b from γ. The $K_{a,b}$ clique can be identical to a maximal complete subgraph or exist on a subset of nodes of a maximal complete subgraph. A $K_{a,b}$ clique community is the union of all $K_{a,b}$ cliques that can be reached from each other through a series of adjacent $K_{a,b}$ cliques. Two $K_{a,b}$ cliques are considered to be adjacent if their overlap is at least a $K_{a-1,b-1}$ biclique. In other words, two cliques must share minimally $a-1$ nodes from δ and also $b-1$ nodes from γ.

Hecking et al. [61] used the algorithm developed by Lehmann and colleagues [76] to identify cohesive subgroups in bipartite graphs of students accessing learning material during a lecture period. The researchers conducted two case studies. One with a group of 44 students attending the lecture on interactive learning and teaching technologies. The other one with a group of 173 students attending an online course on computer-mediated communications. The students had access to learning material such as slides, videos, serious games, as well as a wiki with relevant concepts from the respective lectures. The detected clusters revealed interesting patterns from both groups. A cluster with a high concentration of students was interested in the videos during the lecture time, while other clusters were interested in different learning materials. The researchers tracked the detected clusters until the preparation time before the final exam, discovering that the access to the learning material drastically increased in comparison to the whole semester. This kind of patterns is often studied with a specialized visual metaphor. In the case of dynamic social networks, the best choice are the dynamic graph drawings.

2.5. Visualization of dynamic social networks

Dynamic graph drawings are the metaphors of choice when it comes to the visualization of a dynamic social network. These drawings inherit all the properties of the network under study since they rely on the same representation of time used during the analysis phase. The render window [88] acts as a direct reflection of the time window [60] and defines the period of time in which the network will be illustrated. Dynamic graph drawings created from the continuous representation of time can use the render window to illustrate the complete stream of events or some states at different points in time. In contrast, dynamic graph drawings created from the discrete representation of time can only use the render window to illustrate the successive sequence of states of the dynamic

social network. For each one of them, a graph drawing is computed separately and the resulting sequence is presented to the user in a predefined order. More formally, this drawing is defined as [31]:

$$D(G) = [d(g^1), d(g^2), d(g^3), ..., d(g^n)]$$

Any graph drawing algorithm used to illustrate a static network can be applied to a dynamic social network. These approaches are often integrated into software applications to illustrate more adequately the changes over time.

TecFlow [53] is a software application developed to support the analysis and visualization of e-mailing networks over time. The render window uses the time information of the e-mails to create a dynamic graph drawing in the form of a movie clip. Each frame of the clip has a unique graph drawing, which is computed with the Fruchterman-Reingold algorithm [46]. TecFlow also has time navigation options. It is possible for users to go backward or forward in time and observe the changes occurring in the network. This software application has been used by Gloor [52] to visualize the activities and the communications between the organizers of several conferences.

SoNIA [14] is a software application to visualize dynamic social networks. It uses as the basis the continuous representation of time to create dynamic graph drawings. Although, it is also able to switch to a discrete one. The dynamic graph drawings are presented in the form of a movie clip, where each frame is computed with the Kamada-Kawai algorithm [67]. SoNIA allows users to navigate backward and forward in time in order to observe the changes in the dynamic social network. The appearance or disappearance of actors and their relations are handled with smooth transitions in order to minimize drastic changes from the visual perspective.

In contrast to SoNIA [14], GraphDiaries [7] is based on the discrete representation of time and relies on the Fruchterman-Reingold algorithm [46] to illustrate the dynamic social network. GraphDiaries [7] depicts the changes in such a network with animated transitions. A set of colored halos appears as the users perform a transition between two consecutive periods of time. Blue halos surround the leaving actors. Red halos surround the entering ones, while the remaining elements maintain their position on the canvas.

Knowledge Flow Maps [21] are designed to visualize the evolution of chemistry research over time. They are inspired by the graph drawing techniques but extend their concept with a set of additional features. The nodes represent the different disciplines from chemistry. Each node possesses a pie chart with the number of scientific publications made in the respective disciplines. The edges represent the influence of one discipline over the other and their direction reflects the flow of knowledge at a specific point in time.

Weaver [59] is a software application for visualizing the changes on the members of a community over time and combines two different perspectives for such a purpose. The 2D perspective is used to visualize certain points in time. The actors and relations that appear here are distributed around a circumference as it is stated by the simple space solution algorithm [69]. Since each state of the dynamic network has a different graph drawing, Weaver relies on its 3D perspective to track the actors with ease. This

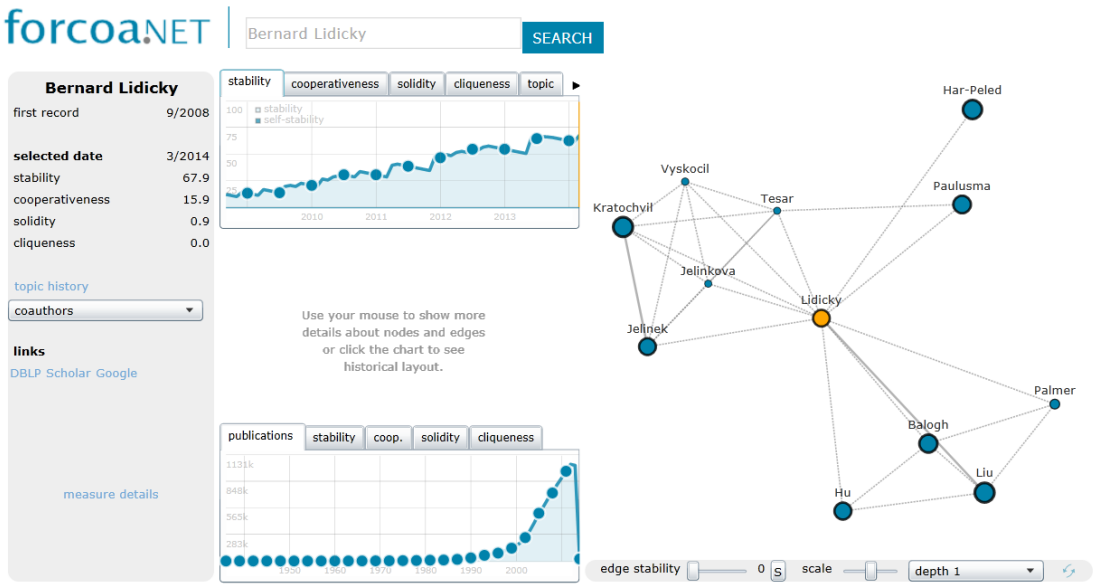

Figure 2.4.: Forcoa.Net analysis tool

perspective displays all the time periods that form the dynamic network. Each one of them is aligned next to each other in chronological order over the Z axis. Furthermore, a set of links highlights the appearance of an actor over time.

Cubix [8] shares some similarities with Weaver [59]. It combines a 3D and a 2D perspective to visualize a dynamic social network. The 3D perspective displays a third-dimensional representation of the adjacency matrix that forms the dynamic social network. The X and Y axis contain the actors, while the Z axis represents time. Such a cube supports interactive operations, like rotation, filtering, and slicing. The slicing operation switches automatically from the 3D perspective to the 2D perspective. Here, the users can observe in detail the actors and relations available at a specific period of time.

Forcoa.Net [62] is a web-based tool for the analysis and visualization of dynamic co-authorship networks. It is mounted on top of the DBLP database from the field of computer science, which contains information about 913,543 authors. Forcoa.Net is based on the discrete representation of time. In addition, it offers a series of features to explore different aspects of a dynamic co-authorship network. On the left-hand side of the user interface, the information panel displays the name of the researcher along with some metrics like stability and cooperativeness [70]. In the center of the user interface, the statistics panel displays statistical visualizations using as the input the metrics aforementioned. Lastly, on the right-hand side of the user interface, the network panel shows the 1.0 neighborhood of the research under observation that existed at a specific point in time. Figure 2.4 illustrates the web-based tool Forcoa.Net.

Twitter Scope [47] is another web-based application for visualizing real-time messages from twitter and it is inspired by the “maps metaphor”. Each text message passes through a semantic analysis process using Latent Dirichlet Allocation method [16] (LDA). LDA assumes that a given text is distributed over one or several topics, which are determined by the set of words. Once the LDA has obtained the topics, the dissimilarity of

the messages is calculated with the Kullbak-Leibler divergence. The analyzed messages are then organized into specific categories. These categories are used to generate a map visualization as next step. Twitter scope assigns a fixed position on the canvas to every category. It is inside of this position where the analyzed messages are displayed. As new messages become available, they replace the old ones by taking their position. Such a characteristic minimizes the changes in the drawing.

Dynamic graph drawings or visual metaphors created with the discrete representation of time have some limitations. As it has been mentioned, a graph drawing is computed separately for each of the states in the dynamic social network. Thus, it is very likely for the actors, relations or other patterns to change their position on the canvas as the users navigate through time. In addition, actors, relations or other patterns can leave the dynamic social network without prior information. Researchers have developed layout adjustment techniques as a solution to this problem. These approaches aim to minimize the difference in the sequence of graph drawings by "minimizing the movement of the layout", "assigning a node to a fixed position" or "reducing the number of elements entering and exiting the visual representation". Such characteristics are mentioned in the theories of visual stability and are related to the origin of the mental map [37].

2.6. Theories of visual stability

In our daily life, the visual system allows us to identify all the objects that are withing the range of vision. A series of fast eye movements, also known as saccades, scan the objects along with their corresponding attributes. The information is then forwarded to the brain, where the image of the world in front of us is generated. An amazing feature of the visual system is that the objects in our sight can be in motion, our body in movement, the eyes scanning and yet, the image of the world appears to be perfectly stable. This is known as visual stability: the perception of a stable world despite the fact that the information in the retina is in continuous motion [78]. Such a property has induced the scientific community to elaborate theories in order to understand how visual stability is achieved [83]. Nonetheless, irrespective of the advancements, visual stability is still considered as one of the real mysteries of visual perception [90].

Von Holst and Mittelstaedt [116] developed one of the first theories of visual stability. According to the researchers, there are signal based mechanisms capable of canceling the continuous movements of the eyes before an image is processed. A signal is emitted from the motor areas of the brain and informs the eyes about their next position on the visual field. This signal is called the "efference copy". Then, the visual system captures the information about the world in our sight, sending it back as another signal. This signal is known as the "afference signal". Before the image can be processed, both signals must pass through mechanisms proposed by Von Holst and Mittelstaedt. The image encoded in the afference signal is shifted by a distance equal to an eye movement. Thus, the mechanism combines the afference signal with the efference copy to cancel the shift. The resulting signal is then forwarded to the brain, where it is perceived as visually stable. This "cancellation theory" was widely accepted among researchers, until further investigations suggested it was part of a more complex system.

MacKay [82] developed an alternative to the previous theory. It considers the existence of a spatiotopic map rather than a mechanism to cancel the shift in the retinal image. According to MacKay, the efference copy is a request of the brain to the visual system to start a comparison between the afference signal and the content of the spatiotopic map. The comparison is performed by an "evaluation mechanism", which has to decide if the displacement of the retinal image is greater than the one calculated for the eye movement. In case the evaluation mechanism detects a difference, the visual system proceeds to update the spatiotopic map. Otherwise, visual stability is achieved. MacKay does not provide a concrete description of the evaluation mechanisms. It is assumed to be flexible and adaptable to different conditions. Although, as it is stated by McConkie [83], it must take into account the spatiotopic map, the afference signal as well as the efference copy to compensate the displacement on the retinal image.

A third theory of visual stability is based on the work of Whipple and Wallach [120]. The researchers conducted a case study to determine the threshold in which a change is perceived in the visual field. A group of participants was asked to perform saccades between two sides of a large circle. The circle was moved to different distances and its displacement was either parallel or orthogonal to the saccades. In addition, Whipple and Wallach used a metric called "displacement ratio". It is defined as the ratio of the displacement size to the saccade length. The results obtained suggested that the displacements in the orthogonal direction must be twice as great as the displacement in the parallel direction to perceive a change. Moreover, the displacement ratio necessary to perceive a change in the orthogonal displacement was greater than the one reported by other researchers [23, 74, 81].

Bridgeman and Stark [24] conducted the same experiment as Whipple and Wallach [120]. The researchers observed that the distance in which the eyes landed on the sides of the circle was greater during a parallel displacement in comparison to an orthogonal one. Such a distance was responsible for the participants to perceive a change. Therefore, the analysis of the data was performed using the distance to the "target-objects" as the basis, e.i., the distance to the edges of the circle rather than the displacement ratio. The results obtained suggested that the distance was, in fact, the factor responsible for perceiving a change in the visual field. Thus, the target-object theory of visual stability states that as long as the distance to the target-objects remains under a certain threshold, the world is perceived as stable.

O'Regan [90] made key assumptions regarding the existing theories of visual stability. At that time, it was known that the brain is not able to store a successive sequence of high-quality panoramic images of the world in our sight during the stabilization process. Likewise, there was no such a long term memory to store all the identified objects in our visual field. Yet, we continue to perceive the world as stable. O'Regan suggested that long-term memory mechanisms were not necessary as all the information required is always in front of our eyes and the visual system can access it at any moment. Therefore, the visual system only needs to relocate certain reference objects to "calibrate" the visual perception.

Bridgeman [25] and Deubel [30] conducted further studies based on the assumptions of O'Regan [90], extending the target-object theory to the "calibration theory" of visual stability. According to Deubel [30], before a saccade is executed, the attention shifts towards a reference object in our visual field. The location along with the attributes

of the reference object are stored in a short term memory known as the "transsaccadic memory". After a saccade, the visual system attempts to relocate the reference objects. Visual stability is achieved if the reference object is found and new objects are identified depending on the distance to the reference object.

The principles mentioned in the theories of visual stability have been used indirectly to develop layout adjustment techniques or layout "stabilization" methods for dynamic graph drawings. These techniques aim to maintain a constant shape [7]; restrict the movement of the nodes and edges [40]; or reduce the number of elements entering and exiting the dynamic graph drawing [31]. It is considered highly relevant in the adjustment or "stabilization" of a dynamic graph drawing, is because minimizing the changes in a sequence of graph drawings helps to preserve the user's mental map [32, 37] and improves the understanding of an evolving network [18, 38, 85, 86].

2.7. Layout adjustment techniques and the mental map

Mental map and the mental distance

The mental map is defined as *"the structural cognitive information a user creates internally by observing the layout of a graph"* [32]. It was introduced in the context of interactive graph drawings, where researchers like Eades [37], Böhringer [18] and Messinger [85] evaluated it for the first time. Nowadays, the scientific community has developed a series of metrics and layout adjustment techniques to study the implications of the mental map in dynamic graph drawings.

For example, Diehl [32] introduced the metric known as the *mental distance*. It quantifies the changes of a dynamic graph drawing to have a better insight about the possible alterations in the mental map. The mental distance is formally defined as follows:

- Let l_1, l_2, $\in$ *Layout* be two graph layouts.
- Let Δ : *Layout* x *Layout* $\rightarrow R_0^+$ be a function to determine how good l_2 preserves the mental map with respect to l_1.
- In case $\Delta(l_1, l_2)$ is equal to 0, both layouts are considered to have the same mental map.

Diehl [32] extended the notion of the mental distance and developed two additional metrics. On the one hand, there is the *euclidean mental distance* which is defined as:

- Let l_1, l_2, $\in$ *Layout* be two graph layouts.
- Let $\Delta_{\|}(l_1, l_2) = \sum_{v \in V_1 \cap V_2} dist(l_1(v), l_2(v))$
- where *dist* refers to the Euclidean Distance of the vertex v in the layout $_l1$ and l_2

On the other hand, there is the *orthogonal mental distance*. It quantifies the difference between two graph layouts using as the basis the horizontal and vertical displacement of the nodes that have in common. The horizontal displacement of a node is defined as the difference on its X coordinate with respect to all the others. Likewise, the vertical displacement of a node is defined as the difference on its Y coordinate with respect to the other nodes. As it is stated by Diehl [32], a "displacement value" must be assigned to the results of each difference. The value of 1 is assigned if there is a positive result. The value of -1 is assigned if there is a negative result. Otherwise, 0 is utilized. Such displacement values must be subtracted one more time and the absolute value of the previous operation represents the displacement of a node. Therefore, the sum of all displacements represents the orthogonal mental distance.

According to Diehl [32], a short orthogonal mental distance suggests that the two graph layouts are similar to each other. On the contrary, a large orthogonal mental distance indicates that there are significant differences between the two graph layouts and could even break the mental map.

Layout adjustment techniques and their characteristics

Most members of the scientific community have been working on layout adjustment techniques, rather than metrics to quantify the changes in dynamic graph drawings. These techniques are organized into three categories based on their characteristics [22].

Linking suggests that all graphs in the sequence are combined into a single graph that has one node for each occurrence of an actor in the dynamic social network. Then, auxiliary edges are created to connect the same actor in consecutive periods of time. A layout of this graph directly yields positions for all nodes in the sequence. The linking method allows the nodes to have a certain freedom of movement. However, the auxiliary edges minimize the displacement of the image similar to the theory of visual stability proposed by Von Holst and Mittelstaedt [116].

Anchoring suggests that the nodes are connected with auxiliary edges to immobile copies fixed to a specific location in the Euclidean Space. The anchoring method reduces the changes in a dynamic graph drawing similar to the target-object theory of visual stability proposed by Bridgeman and Stark [24].

Aggregation suggest that all nodes and edges appearing in a successive sequence of graphs must be aggregated into a single graph. This graph exemplifies all the occurrences of an actor or entity with a single node and applies the same principle to the relations. The position of these elements in the Euclidean Space is determined by applying a graph drawing algorithm to the aggregated graph.

The foresighted graph layout algorithm

Diehl [31] developed an algorithm to create dynamic graph drawings based on the aggregation method. The approach is called the *Foresighted Graph Layout* (FGL) because it knows the future of the given graph. The FGL relies on three main concepts. First,

a *partition* refers to a collection of disjoint sets, whose union is equal to the original set. Second, a *component* refers to a single element in the collection of disjoint sets, whose intersection between any other component is equal to an empty set. Lastly, there is the *lifetime* which refers to an attribute describing the appearance of an entity or relation in a dynamic network.

The concepts described beforehand are used by the foresighted graph layout algorithm as follows. All entities and relations appearing in a dynamic network are aggregated into a single graph. This *Super Graph* exemplifies all occurrences of an entity with a single node. Likewise, all occurrences of a relation are represented with a single edge. According to Diehl [31], the super graph requires a considerable amount of space to visualize a dynamic network and "compressing" it might solve such a limitation.

The *Graph Animation Partitioning* (GAP) executes a compression process on the node set of the super graph. A set of "containers" called *node components* are used to store the entities appearing in the dynamic network. The entities are allowed to share a position in the same node component as long as they do not appear in the same period of time. In order words, they must have disjoint lifetimes. As a result, the GAP returns the node partition of the foresighted graph layout.

As next step, the *Reduced Graph Animation Partitioning* (rGAP) executes a compression process on the edge set of the super graph. A set of "containers" are used to store the relations appearing in the dynamic network. These containers are called *edge components* and operate exactly like their node counterparts. All relations that do not appear in the same period of time are allowed to share a position in the same edge component. More formally, the relations must have a disjoint lifetime. As a result, the rGAP returns the edge partition of the foresighted graph layout.

The partitions obtained from the compression process form the foresighted graph layout, which can be drawn in the Euclidean Space with any graph drawing algorithm. Furthermore, such a drawing presents two interesting characteristics. On the one hand, the entities and relations of the dynamic network will always appear in the same node or edge position respectively. Still, a position can contain multiple entities or relations at different points in time. On the other hand, the drawing possesses the minimal number of changes in its structure during the exploration of the dynamic network. Figures 2.5 and 2.6 show the algorithm to compute the Foresighted Graph Layout, while Figure 2.7 provides an example of a dynamic graph drawing computed with such an algorithm.

Empirical findings

As more layout adjustment techniques are introduced to the field, other researchers decided to evaluate their efficiency through case studies. Purchase [99] was the first one to conduct a case study about the stability of a dynamic graph drawing. The dynamic network used in this study was the structure of an imaginary website over six periods of time. The nodes represented web pages, while the edges the links between them. Three different configurations of such a network were used in the case study. Each one of them contained around 30 nodes and 30 edges. In addition, each configuration had 4 changes in each period of time. The changes were related to the addition or removal of a node or an edge.

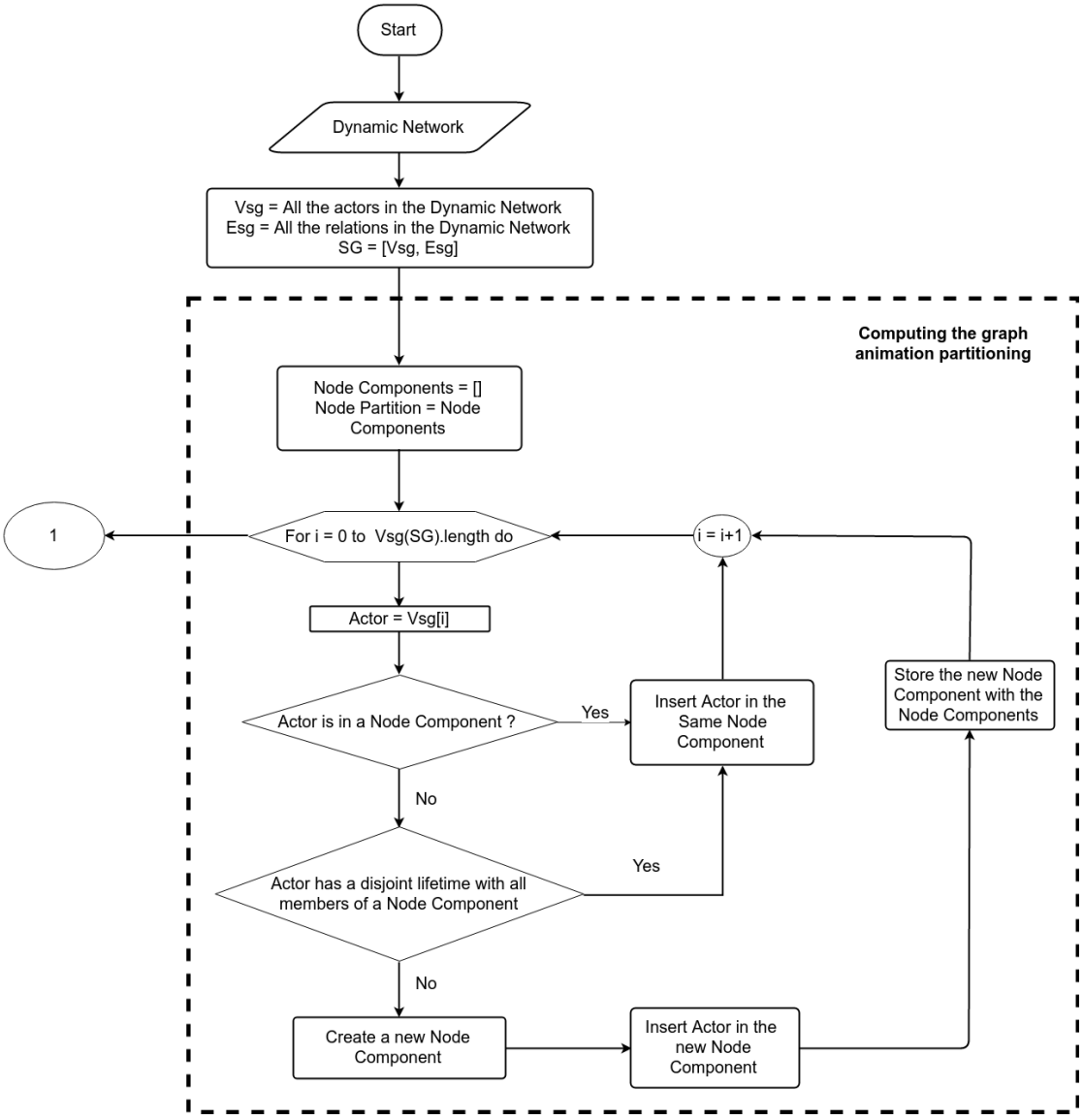

Figure 2.5.: Algorithm to compute the foresighted graph layout - part 1

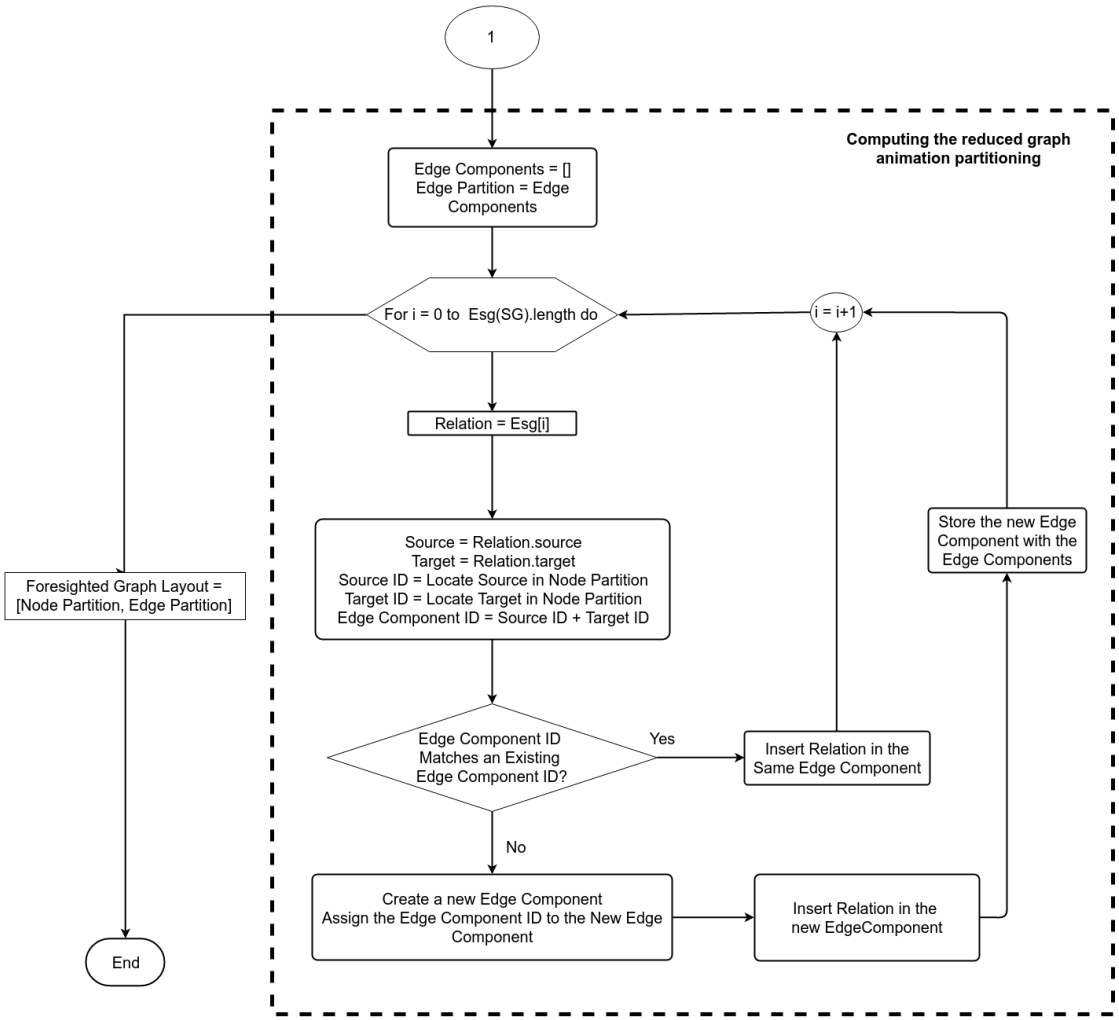

Figure 2.6.: Algorithm to compute the foresighted graph layout - part 2

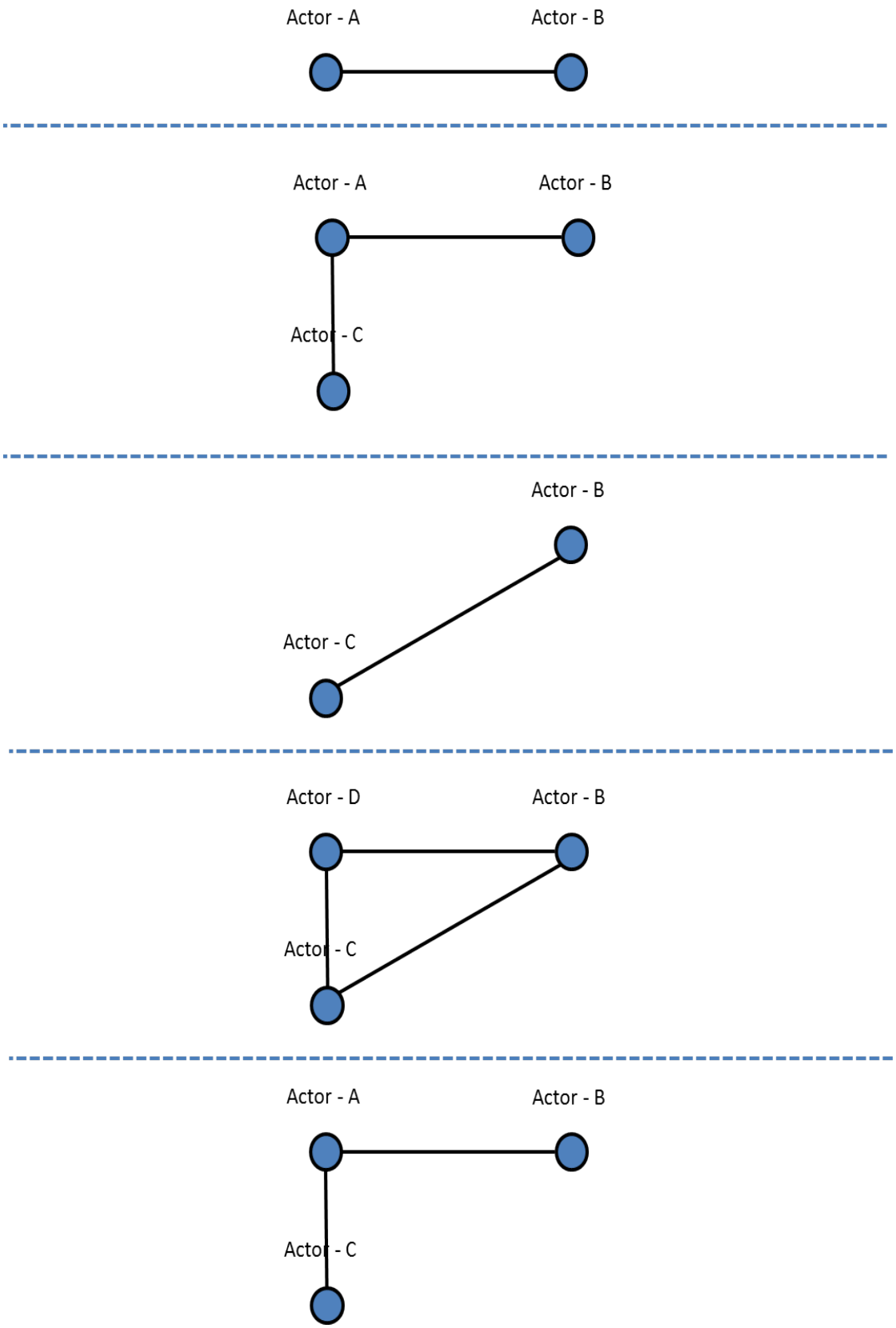

Figure 2.7.: Visualizing a dynamic network with the Foresighted Graph Layout

A dynamic graph drawing was created for each configuration using the hierarchical layout available in *GraphAnimation* [32] and stabilized with the same tool. GraphAnimation stabilizes the layout using a value as input for the parameter *delta*. A higher value of delta allows the nodes and edges more freedom of movement, whereas a lower value of delta has the opposite effect. Therefore, the three drawings have three different conditions of stability. One with delta = 0 (stable), another with delta = 20 (semi-stable) and the last one with delta = 40 (unstable).

A group of 20 participants from the fields of computer science, zoology, music, and law participated in the case study. The students were asked to complete a series of tasks in the different dynamic graph drawings. The tasks focused on the identification of new links added to the website, the page that changed the most over the years, the year in which the site reduced its size and the year in which a page became available through another page. By the end of the tasks, the users were requested to answer a questionnaire about the changes in the network. The analysis of the data involved the response time of the users in the respective tasks and the errors produced in the questionnaires. The results obtained from the study suggest that the stability of a dynamic graph drawing improves the identification of nodes by their names. However, it is less important for tasks related to the edges.

In a subsequent study, Purchase [98] evaluated the aesthetic criteria and the stability of a dynamic graph drawing. The study presented three different dynamic networks. Each one represented telephone communications between a group of students over six periods of time. The three networks had 22 nodes and 33 edges, with approximately 22 changes regarding the edges. No new nodes were inserted into the network and the existing ones were never removed.

The dynamic graph drawings were created with *GraphAEL* [40] and stabilized using the same tool. A set of auxiliary edges interconnected the same node in two subsequent periods of time. These edges had a strength level which was specified by Purchase to prevent the movement of the nodes in the dynamic graph drawing. Three different configurations were suggested. One with a strength level of 40 (stable), another with a strength level of 5 (semi-stable) and the last one with a strength level of 0.1 (unstable).

A group of 30 students participated in the study. The students were asked to complete a series of tasks in the dynamic graph drawings with their respective conditions of stability. By the end of the tasks, the students were asked to answer a questionnaire related to the connectivity of the nodes in the dynamic network. The collected data was analyzed taking into account the response time of the users in tasks and the errors produced in the questionnaires. The results suggested that the unstable layout and the stable layout have a better performance in the identification of nodes with high connectivity. The semi-stable layout had the worst performance during the evaluation, producing more errors and longer response times. Thus, Purchase [98] argued that it was not possible for the aesthetic criteria and dynamic stability to coexist in the same dynamic graph drawing. This was corroborated by Ghani [51].

In a more recent study, Archambault [4] evaluated the stability of a dynamic graph drawing and its relationship with the memorability of the dynamic network. Memorability measures to which extent a person remembers an object in the information space [51]. Three different dynamic networks were used in the study. Each one of them had six periods of time. The dynamic graph drawings were created with GraphAEL [40] and

stabilized using the same tool. Three conditions of stability were applied to the dynamic graph drawing of each network. An unstable drawing with a linking strength of 0.1, a semi-stable drawing with a linking strength of 60 and a stable drawing with a linking strength of 400.

A group of 25 students participated in the study. They were asked to complete a series of tasks divided into two phases. The first phase requested the users to memorize three dynamic graph drawings, whereas the second phase displayed three graph drawings to the students. Some drawings appeared in the first phase but others were completely different. The students were requested to identify those dynamic graph drawings that were used in the first phase and also those ones which were different. The collected data was analyzed with respect to the response time of the users and the errors produced during the identification of the dynamic graph drawings. The results of the study suggested that the stability of a dynamic graph drawing improves the memorability of a dynamic network.

2.8. Summary

This chapter presented the state of the art about the analysis and visualization of social networks. Social networks analysis techniques have allowed researchers to study a variety of real world scenarios such as the identification of cohesive subgroups, the time it takes a person to be infected by a disease and also the organization of entire urban areas. These analyses have been complemented using graph drawings in order to have a better understanding of the discovered pattern. The analysis and visualization of a social network only represent the state of this structure at certain points in time.

Dynamic social networks were introduced to study actors and their relations at different points in time. The analysis techniques for these networks have allowed the identification of researchers in co-authorship networks, cohesive subgroups in learning scenarios and also other patterns. These results were visualized using dynamic graph drawings. Since the analysis of a network is often conducted following the discrete representation of time, dynamic graph drawings must use the same approach. For each snapshot in the dynamic network, a graph drawing is computed separately with the algorithm of preference and the resulting sequence is presented to the user in a predefined order. Despite the simplicity of the approach, dynamic graph drawings created with the discrete representation of time cause some issues. Actors and their relations appear in different positions over the drawing area during the exploration of the dynamic social network. Moreover, it is very likely for an actor, relation or pattern to disappear without prior information. The combination of these factors can disturb the user to the point of breaking the mental map.

The preservation of the mental map has become one of the highest priorities in the visualization of dynamic social networks. It has been reported that the mental map helps to improve the understanding of an evolving network. For this purpose, a series of layout adjustment techniques have been developed. Aggregation techniques aggregate all the nodes and edges in the dynamic network into a single graph, in which the position of each node is determined by the layout of the aggregated graph. Linking techniques combine all graphs in the sequence into a single graph, in which there is a node for each

appearance of an entity over time. Auxiliary edges are created to connect the appearance of same nodes in subsequent periods of time. These edges prevent the movement of the nodes on the canvas. Anchoring techniques connect the nodes to a fixed position in the Euclidean Space. The layout adjustment techniques present characteristics related to the principles of visual stability and can be used to study dynamic graph drawings from a different perspective. In the subsequent chapter of this thesis, a framework to support flexible visualization techniques is presented. It served as the basis to develop dynamic graph drawings and layout adjustment techniques. These approaches were evaluated through a series of case studies emphasizing on the user experience and the efficiency of the visual search. As part of such an evaluation, a mathematical of visual stability was used to characterize those drawings supporting the attributes aforementioned in a dynamic scenario.

3. A framework to support flexible visualization techniques

3.1. Introduction

This chapter presents the framework to support flexible visualization techniques that were developed in the European Union project SISOB. It addresses the technical limitations of the software applications to visualize dynamic social networks described in Chapter 1. On the one hand, the framework eases the integration of graph drawing algorithms, layout adjustment techniques, and specialized visual metaphors into the same analysis platform, the web-based analytics workbench [54]. On the other hand, the drawing engine introduced by the framework uses the D3.js[1] library to create visual representations based on scalable vector graphics. This last component can utilize the multi-agent system of the workbench to process the drawing server-side, client-side or by combining both strategies in case it is necessary. The framework to support flexible visualization techniques served as a common platform to study the implications of visual stability in dynamic graph drawings.

3.2. The web-based analytics workbench

The web-based analytics workbench [54] is a software application that was also developed during the run of the SISOB project. It is designed to create and execute network analysis workflows. The workbench introduces a web-based user interface inspired by the pipes-and-filters metaphors. Each filter represents a social network analysis technique, which can be interconnected with others to perform complex analysis operations. In addition, the workbench counts with a server-side computational backend to execute the actual analysis. A Node.js server[2] hosts the user interface along with a repository to store the analysis results. Moreover, it provides a communication channel from the user interface to an open source implementation of the *TupleSpace* concept [50] used by a multi-agent system, the *SQLSpaces* [119].

The execution of an analysis workflow sends the information of the corresponding filters, along with the input data, to the SQLSpaces. Once the information has reached this location, the multi-agent system instantiates the agents that will be involved in the analysis process. An agent fetches the input data from the SQLSpaces, processes it and sends the results back to the same location where the next agent can find it. In some situations, the agents are allowed to fetch and return the results to a different location. Input agents or "data sources" normally acquire the data from a local computer or from an external data repository. Output agents or "data sinks" usually write the data to the

[1] `http://d3js.org/`
[2] `https://nodejs.org/`

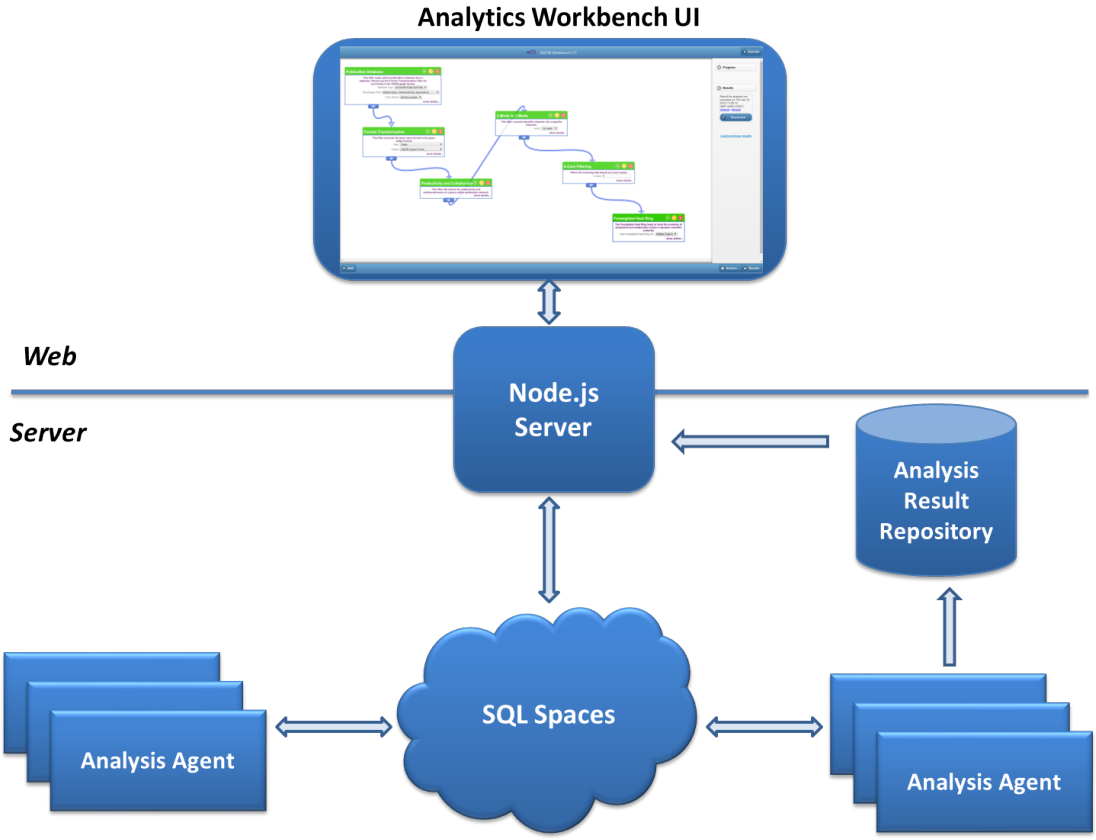

Figure 3.1.: Architecture of the web-based analytics workbench

repository where the analysis results are stored. The results of an analysis workflow are offered to the outside through the Node.js server. Figure 3.1 illustrates the architecture of the web-based analytics workbench.

As for the underlying graph representation, the web-based analytics workbench uses a "decoration" approach. This means, when an agent completes its routine, it decorates the data with the results obtained (i.e., it superimposes the results typically as attributes over the original graph). This feature has two main benefits. On the one hand, it makes the ordering of the filters more flexible since the original information is never destroyed but only extended. On the other hand, the main type of data analyzed with the workbench are networks. This incremental decoration approach makes possible to preserve the context of the data at any time, even though the analysis emphasizes on the nodes or the edges. Thus, output filters like web-based visualization techniques can access the original network along with the attributes calculated in every step of the analysis.

The web-based analytics workbench has its own format to encode network data. It was developed using a JavaScript Object Notation (JSON) to ease the integration of web-based visualization techniques. The format, also known as *SISOB Graph Format*, presents two main segments: *metadata* and *data*. The data segment contains all the information about the nodes and edges of the network under study. Both types of entities are stored in their corresponding segments. The *nodes* segment stores node

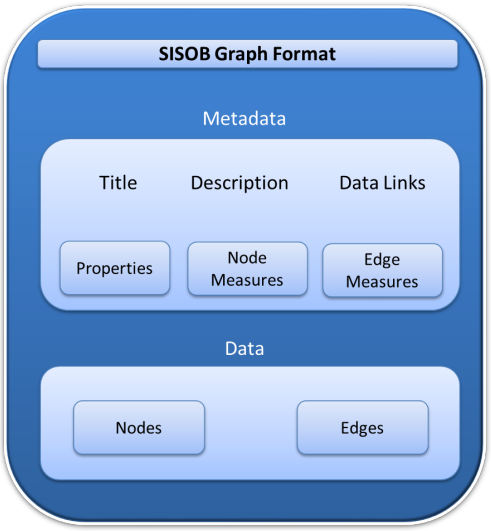

Figure 3.2.: Overview of the SISOB Graph Format

objects with an *id* and a *label* as mandatory fields. The *edges* segment stores edge objects with an *id*, a *source* and a *target* as mandatory fields. The source and the target of an edge object refer to the ids of the pair of node objects that are connected. As the agents perform the analysis, additional attributes decorate the elements in the data segment and a descriptive information is included in the metadata segment. In the case of numeric attributes, the descriptive information is stored under the *node measures* or *edge measure* sub-segments. In the case of non-numeric attributes, the information is stored under the *properties* sub-segment. A sub-segment to encode dynamic networks is in available in the metadata segment. The *datalinks* store the names of the files that form the successive sequence of network snapshots. Lastly, it is possible to include a *description* of the network and a *title*. Figure 3.2 depicts the SISOB Graph Format.

As a complementary feature, the workbench encodes statistical information in a table format. It was developed following the same scheme as the network format. The *SISOB Data Table Format* presents two main segments as well: *metadata* and *data*. Here, the data segment is used to store *data fields* rather than nodes and edges. A data field, requires an *id* and a *label* as mandatory fields. The metadata segment is used to store descriptive information about the attributes calculated during the analysis process. In the case of numeric attributes, the information is stored under the *measures* sub-segment. The non-numeric attributes are not supported in the data table format. Nonetheless, the metadata segment supports *datalinks*, *description* and *title* sub-segments. Figure 3.3 displays the SISOB Data Table Format.

In spite of having its own formats to encode network and statistical information, the workbench supports other data schemes as input or output to the analysis process. These schemes can be transformed into the respective internal formats and vice versa using a format transformation agent. Among the data schemes to encode networks the workbench is compatible with Pajek.net[3], UCINET DL[4] and GML[5]. Among the

[3] `http://mrvar.fdv.uni-lj.si/pajek/`
[4] `https://sites.google.com/site/ucinetsoftware/home`
[5] `http://www.fim.uni-passau.de/fileadmin/files/lehrstuhl/brandenburg/projekte/gml/`

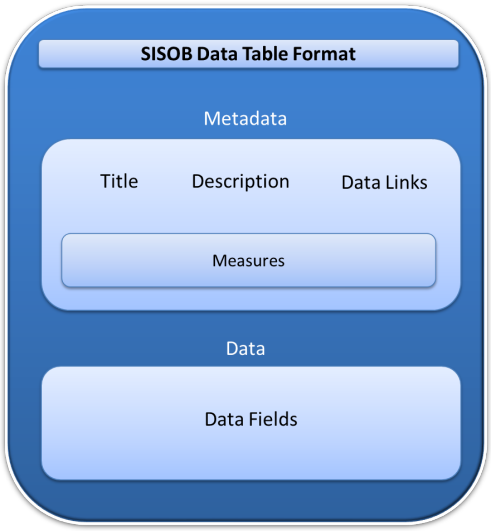

Figure 3.3.: Overview of the SISOB Data Table Format

schemes to encode statistical information the workbench is compatible with CSV[6] as data table format, while JSON Activity Streams[7] is used for log data. The features of the workbench were used as the basis to develop the framework to support flexible visualization techniques.

3.3. A framework to support flexible visualization techniques

The framework to support flexible visualization techniques was created to complement the analysis functions of the web-based analytics workbench. Its user interface, along with the drawing engine, were developed using exclusively web technologies. In addition, the framework uses the architecture of the workbench to introduce its own server-side computational backend. This component relies on the execution of *visualization agents* to process the visual metaphor and to display the available techniques as visualization filters. The visualization filters can be interconnected with the analysis filter due to the use of the same internal formats. Furthermore, the server-side computational backend counts with a *visualization repository* that allows the deployment, update or removal of any web-based visualization technique without affecting the web-based analytics workbench. Figure 3.4, shows the architecture framework to support flexible visualization techniques.

User interface

The user interface acts as a unified viewport to the implemented visualizations, where the actual drawing is displayed. This component depends on three additional modules

gml-technical-report.pdf
[6] http://www.ietf.org/rfc/rfc4180.txt
[7] http://activitystrea.ms/specs/json/1.0/

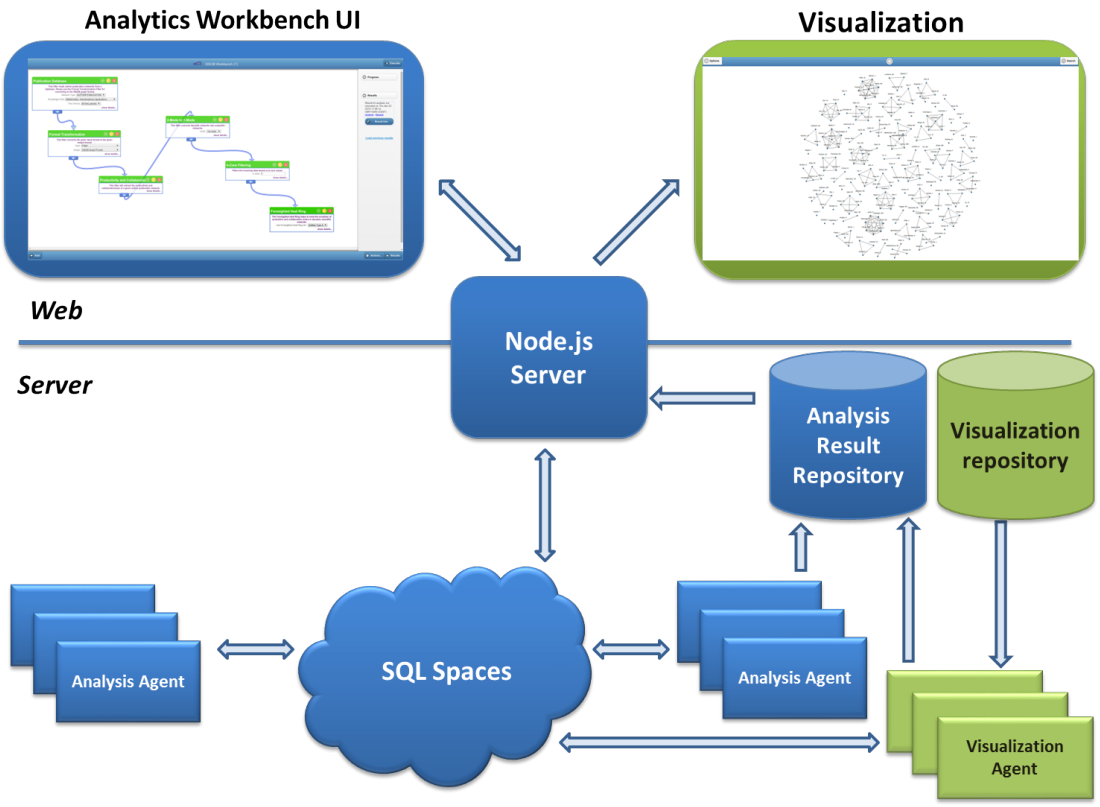

Figure 3.4.: Architecture of the framework to support flexible visualization techniques and its integration with the web-based analytics workbench

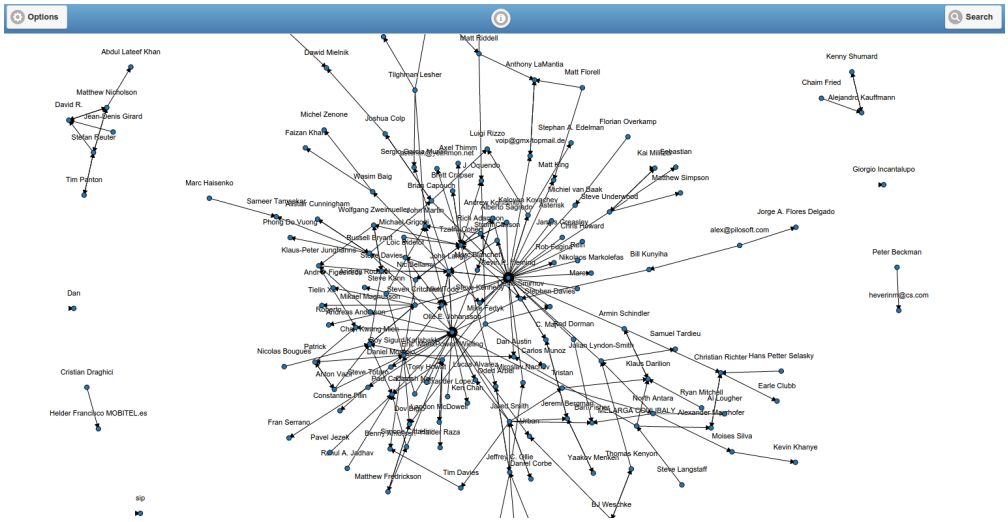

Figure 3.5.: User interface of the framework to support flexible visualization techniques

in order to have an appropriate execution. The *index.html* uses jQuery[8] mobile to create the controls of the user interface. The *framework.js* provides functionality to the user interface. Lastly, the *engine.js* produces the actual drawing. The engine uses internally the D3.js[9] library to create drawings based on scalable vector graphics (SVGs). The web technologies of the user interface allow the visualization techniques to be executed on any kind of platform, including mobile devices. Figure 3.5 illustrates the user interface of the framework.

Visualization agents

The server-side computational backend introduces the *visualization agents*; a specialized version of the agents employed by the web-based analytics workbench. The visualization agents access and process information like any other agent. However, their objective is to assist the drawing engine on the user interface to process either the actual drawing partially or completely. The framework possesses three types of visualization agents: *client-side visualization agents*, *server-side visualization agents* and *hybrid visualization agents*. Each one distributes the processing of the drawing in a different way.

A *client-side visualization agent* fetches the analyzed data from the SQLSpaces. Afterward, it looks for the modules of the web-based visualization in the visualization repository. The analyzed data, along with the modules, are deployed to the analysis result repository of the web-based analytics workbench. This visualization agent does not perform any operation regarding the processing of the drawing. Rather than that, it delegates this task to the drawing engine on the user interface and it is suitable for drawings with low complexity.

In contrast, a *server-side visualization agent* fetches the analyzed data from the SQLSpaces and processes the complete drawing on the server. The analyzed data is decorated with the coordinates or attributes required by the drawing engine. Then, the visualization

[8] `https://jquerymobile.com/`
[9] `http://d3js.org/`

agent looks for the modules of the web-based visualization in the visualization repository. The decorated data, along with the modules, are deployed to the analysis result repository of the web-based analytics workbench. In this scenario, the decorated data is loaded by the modules of the web-based visualization and the engine on the user interface only reads the coordinates or attributes to create the drawing. Server-side visualization agents are suitable for drawings with high complexity.

On the other hand, a *hybrid visualization agent* combines the characteristics of the visualization agents aforementioned. The analyzed data is fetched from the SQLSpaces and the complex operations of the drawing are performed on the server. The resulting coordinates or attributes are used to decorate the analyzed data. Afterward, the visualization agent looks for the modules of the web-based visualization in the visualization repository. The decorated data, along with the modules, are deployed to the analysis result repository of the web-based analytics workbench. In this scenario, the decorated data is loaded by the modules of the web-based visualization. The engine accesses the attributes or coordinates calculated on the server and performs less complex operations to complete the drawings. Hybrid visualization agents are suitable for drawings that involve complex and simple operations.

Visualization repository

The visualization repository is a location on the server, where the modules of the web-based visualization techniques are stored. Each technique requires a folder with a representative name. Inside the folder, it is necessary to include the *index.html*, the *framework.js* and the *engine.js*. Such characteristics are mandatory since the visualization agents access the repository in order to get the respective modules. Despite being on the same server, the visualization repository does not interfere with the configuration of the web-based analytics workbench. As a result, it is possible to update, deploy or remove visualization techniques at any time.

Features of the framework

The framework to support flexible visualization techniques is equipped with a set of features to ease the exploration of networks, including their dynamic variations. The features take into account the attributes calculated during the execution of an analysis workflow. The details are presented below.

Search

The search function receives a keyword as input and looks for all those nodes whose labels matches the given parameter. The results are highlighted in the graph drawing, while the rest of the elements are hidden from the user. In addition, there are available search scopes to support the interactive exploration of the network [58]. As the user selects a node, its neighborhood is highlighted according to the selected scope [58]. The *1.0 scope* covers the selected node along with its immediate neighborhood. The *1.5 scope* covers the selected node, its immediate neighborhood and the neighbors of these last nodes. The *2.0 scope* operates as the scope 1.5, but goes one level further in the neighborhood

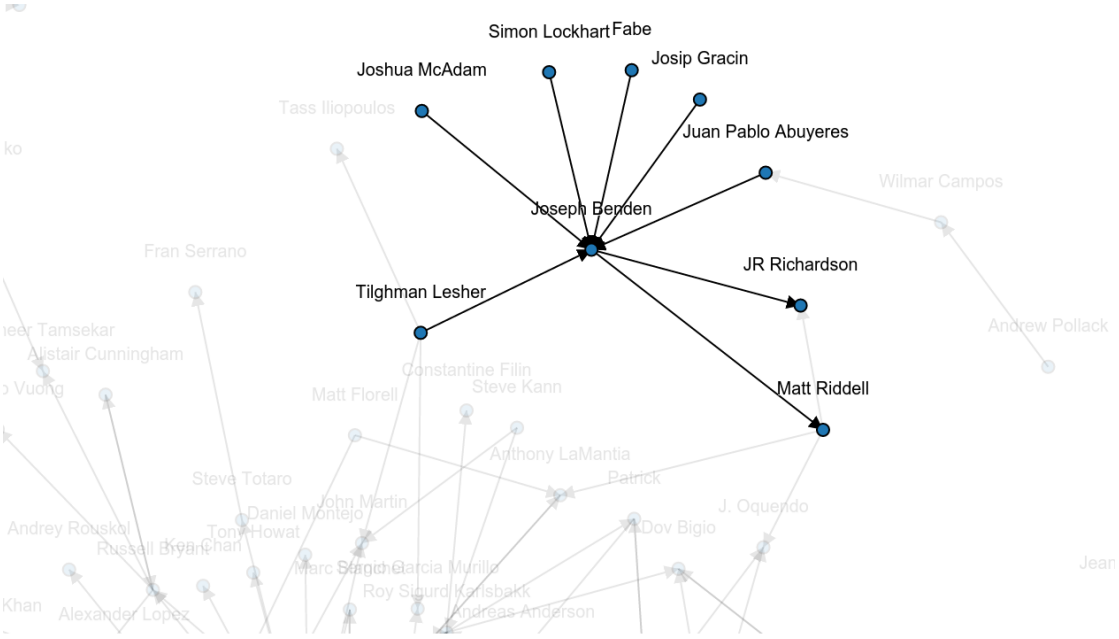

Figure 3.6.: Highlighting the neighborhood of an actor with the 1.0 scope

of the selected node. As an alternative to explore multiple nodes at the same time, the framework introduces the trail scope. This scope covers the same neighborhood as the 1.0 scope. Nevertheless, it always maintains visible the results of the previous exploration. As a result, multiple neighborhoods can be observed in the same drawing. Figures 3.6, 3.7, 3.8 illustrate the neighborhood of an actor with the different search scopes, while Figure 3.9 illustrates the neighborhood of multiple actors with the trail scope.

Tooltips

The framework to support flexible visualization techniques includes tooltips to provide additional information about the analyzed nodes. The information is displayed as soon as the user moves the mouse on a node. Furthermore, it is possible to select multiple attributes of the nodes from the user interface to create custom tooltips with supplementary information.

Styles

Another feature of the framework is the styling of the elements appearing in the graph drawing. The scaling function handles the base scale of the labels, nodes, and edges. Each element has its own base scale and can be adjusted individually to improve the readability of the graph drawing [35]. Increasing the values of the base scales leads to a graph drawing with labels of bigger size, nodes with bigger radius and thicker edges. Contrarily, decreasing the values on the base scales leads to a graph drawing with labels of smaller size, nodes with shorter radius and thinner edges.

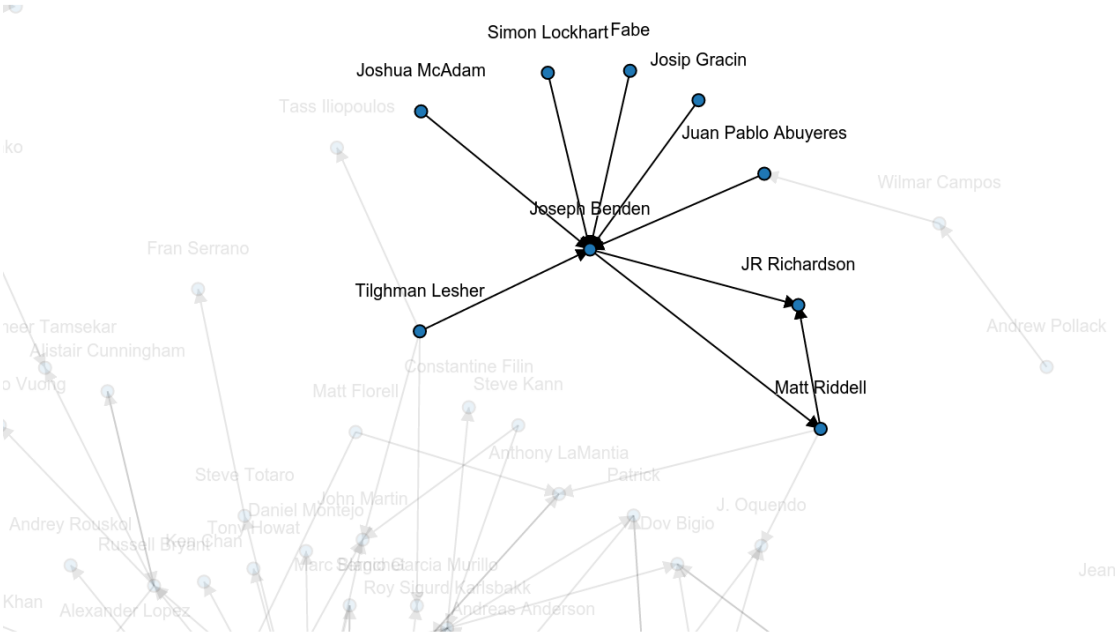

Figure 3.7.: Highlighting the neighborhood of an actor with the 1.5 scope

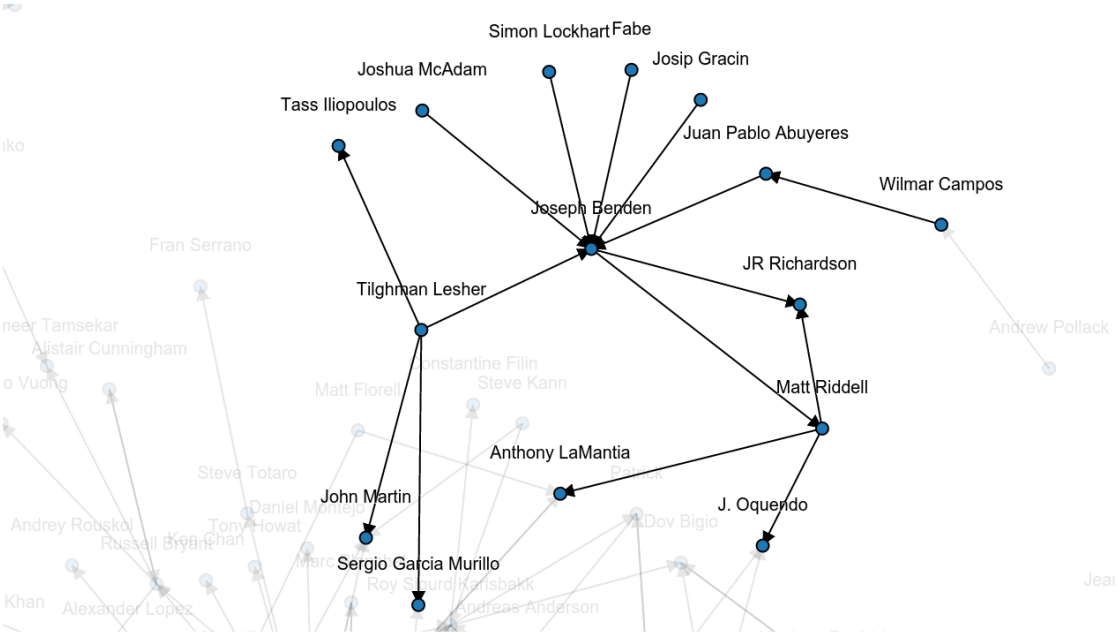

Figure 3.8.: Highlighting the neighborhood of an actor with the 2.0 scope

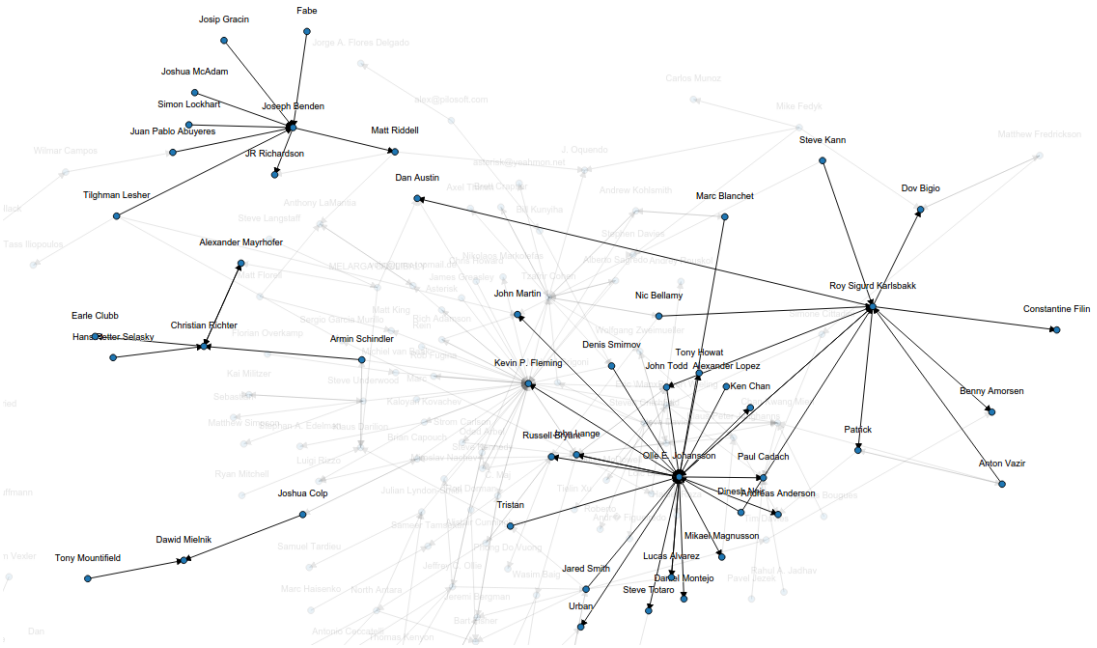

Figure 3.9.: Highlighting multiple actors and their immediate neighborhoods with the trail scope

The mapping function of the framework accesses the attributes decorating the analyzed graph to change the visual appearance of the labels, nodes, and edges in the drawing. For instance, it is possible for the labels to display the numeric or non-numeric attributes attached to the nodes. This feature is useful to visualize the attributes calculated by the web-based analytics workbench.

The nodes have a different behavior in comparison to the labels. The mapping function uses the numeric attributes as a scaling factor and applies it to the base scale of these elements. As a result, nodes with higher values present larger radius, while nodes with lower values present the opposite [19–21]. The mapping function also supports color schemes. In the case of numeric attributes, it selects a color for the minimum value and another one for the maximum. Both are used to create a color gradient, which is applied to the nodes in the graph drawing [39, 125]. In the case of non-numeric attributes, the mapping function asks the users for the specific colors or offers the possibility to show them as geometric shapes.

The edges operate in a similar way. The mapping function uses the numeric attributes as a scaling factor, which is applied to the base scale of these elements. As a result, edges with higher values appear thicker in comparison to the others [19–21]. The mapping function also supports color schemes for the edges. In the case of numeric attributes, it creates a color gradient, whereas in the case of non-numeric attributes it asks the users for the color coding.

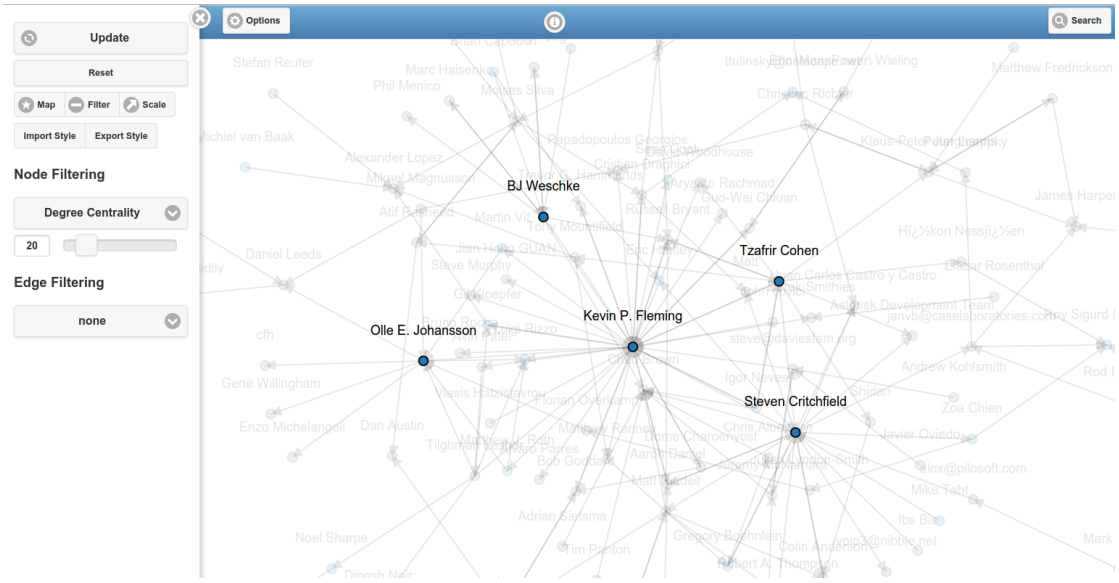

Figure 3.10.: Filtering nodes based on their degree centrality

Filtering

Networks with a considerable amount of nodes and edges tend to hide relevant information from the user. In such scenarios, the filtering function can remove the unnecessary elements from the screen. The attributes decorating the nodes and edges are used as the basis for this operation [28]. In the case of numeric attributes, all those elements whose value is below a user-specified threshold are removed. In the case of the non-numeric attributes, all those elements matching a user-specified criteria are removed. Figure 3.10 depicts the use of the filtering function.

Dynamic networks

The web-based analytics workbench uses the discrete representation of time to create dynamic networks. The time window defines the period of time in which a given network will be under observation [60]. It is utilized to capture a successive sequence of snapshots from the network timeline, which is later encoded with SISOB Graph Format. This information is used by the framework to support flexible visualization techniques to produce dynamic graph drawings. For each of the encoded snapshots, a graph drawing is computed with the algorithm of choice and the resulting sequence is presented to the user in a predefined order [14].

The time navigation function offers two modes to explore a dynamic network. The non-automatic mode allows the users to switch manually between consecutive periods of time. In addition, it is possible to introduce the index of a specific snapshot and visualize it immediately. The automatic mode allows the user to observe the sequence of snapshots as a movie clip. The framework uses a delay to handle the transitions between consecutive periods of time. Furthermore, it is possible for the user to specify such a delay in order to slowly observe the dynamic network. Figure 3.11 displays the time navigation function.

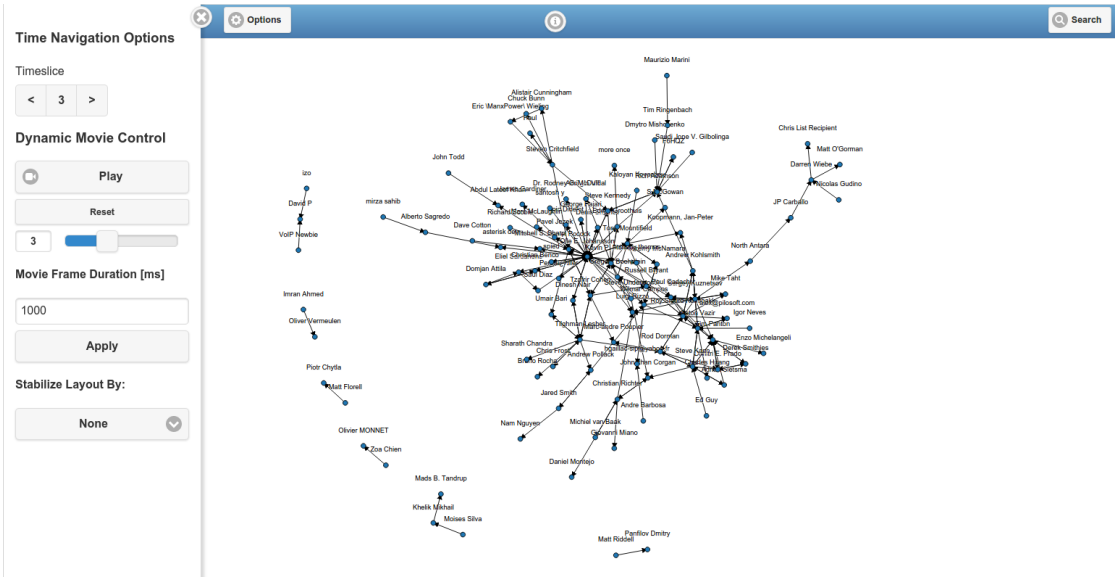

Figure 3.11.: Using the time navigation function to explore the third snapshots of a dynamic graph

The framework also offers basic layout adjustment techniques, although, more sophisticated algorithms can be integrated through the visualization agents. The graph structure technique maintains as much as possible the "shape" of the graph drawing during the time navigation. The drawing engine collects the coordinates of nodes that are available in the current drawing and transfers them either to the next graph drawing or to the one specified by the user. The fixed position technique maintains the position of the nodes during the time navigation based on two distinct criteria. On the one hand, the label criterion keeps the nodes in the same position as long as the same label appears in a subsequent period of time. On the other hand, the id criterion keeps the nodes in the same position as long as the same id appears once again.

3.4. Implemented visualization techniques

A series of flexible visualization techniques were implemented using the features provided by the framework. The notion of flexible comes from adding, updating or removing visualization techniques without affecting the web-based analytics workbench and from selecting the behavior of the drawing engine, depending on the complexity of the visual metaphor. The implemented visualizations are presented next.

Force-directed graph layout

A force-directed graph layout is available on the D3.js library. The approach is based on the use of constraints [36] in combination with physically-based modeling [66]. A repulsion force is applied to all the nodes that are disconnected. At the same time, a pseudo-gravity force maintains the nodes centered in a visible area and prevents the components of the graph to move away. Furthermore, the edges receive a fixed-distance

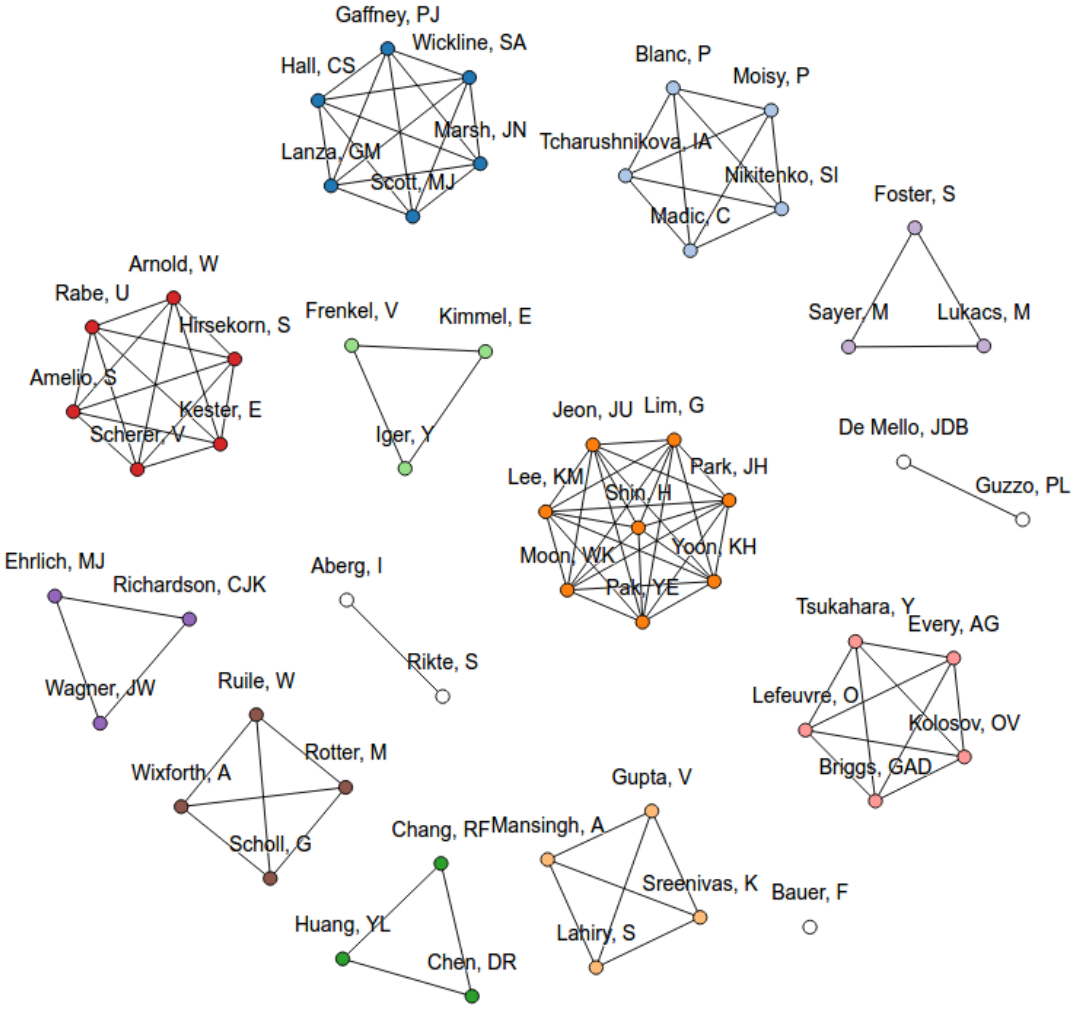

Figure 3.12.: Force-directed graph layout

geometric constraint. The force-directed graph layout continues to adjust the drawing until the edges have reached the specified length. If necessary, the algorithm allows the addition of custom forces or constraints to improve the quality of the drawing. The force-directed graph layout was integrated into the web-based analytics workbench using a client-side visualization agent. Figure 3.12 illustrates the force-directed graph layout.

Force-directed graph layout enhanced with convex hulls

The force-directed graph layout was enhanced to include the use of convex hulls. These hulls are designed to cover members of the same cluster, cohesive subgroup or community and make use of colors to highlight them. In some situations, community detection algorithms identify actors that belong to more than one cluster [92]. The force-directed graph enhanced with convex hulls displays such actors with a black color, indicating an overlap between multiple clusters. Moreover, the color of the convex hulls can be adjusted using the styling function of the framework. This enhanced version of the

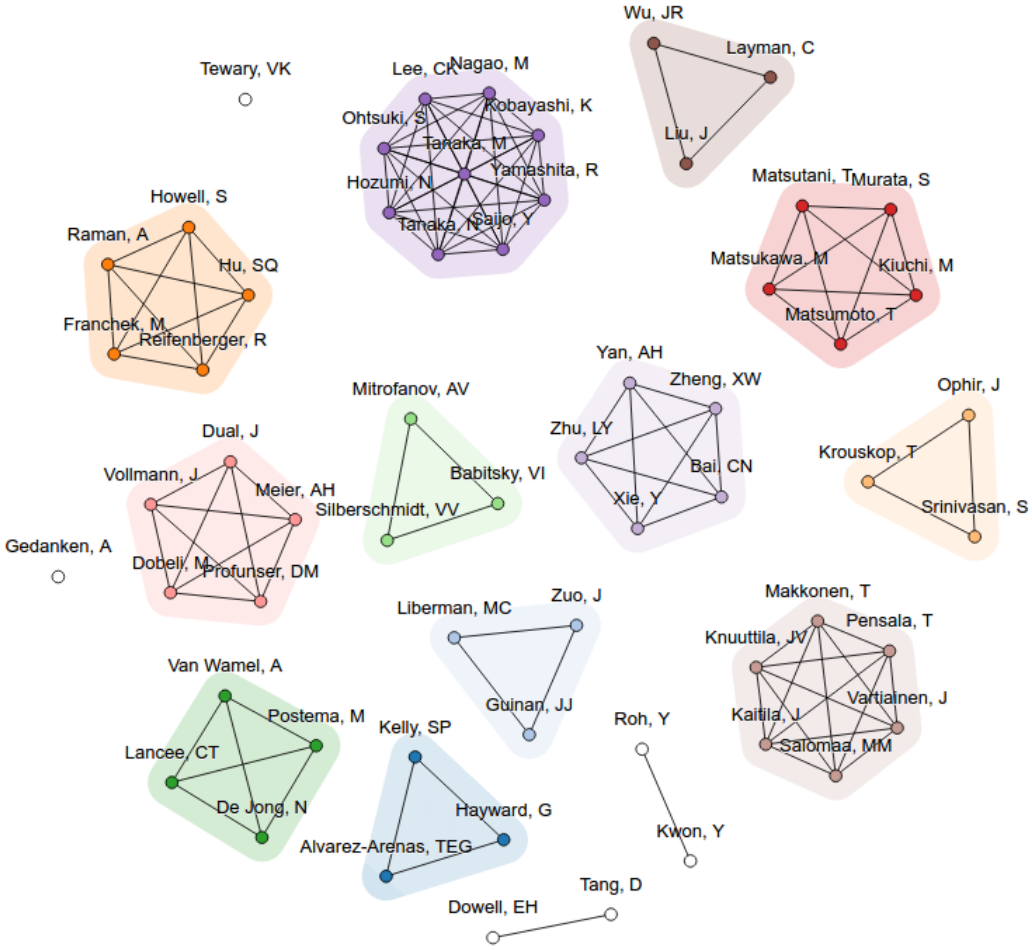

Figure 3.13.: Force-directed graph layout enhanced with convex hulls

force-directed graph layout was also integrated into the workbench using a client-side visualization agent. Figure 3.13 presents the force-directed graph layout with convex hulls.

Circular layout

The circular layout [48] is a visualization technique that distributes the nodes of the graph over a circumference, while the edges appear in its center. Due to its simplicity, the circular layout was integrated into the workbench using a client-side visualization agent. Figure 3.14 displays the circular layout.

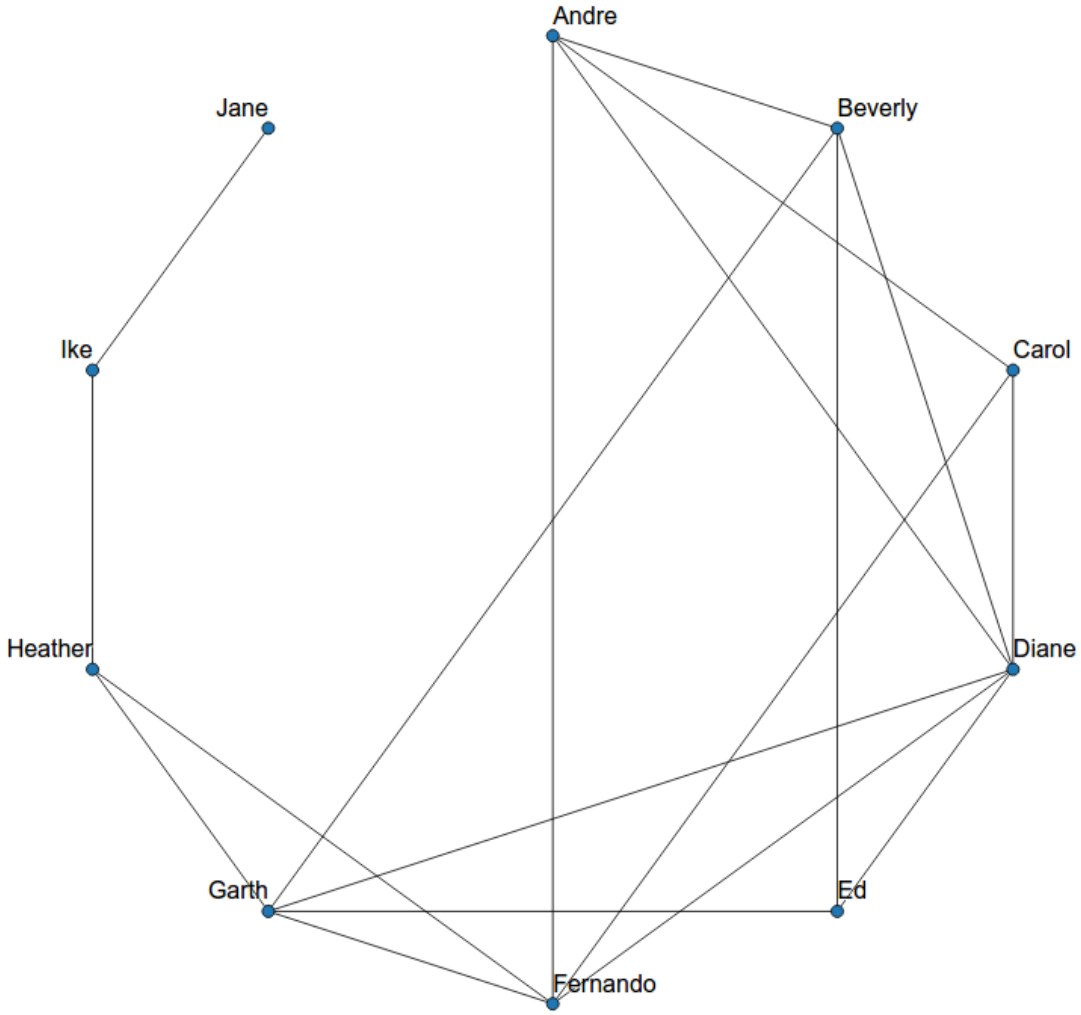

Figure 3.14.: Circular Layout

Fruchterman-Reingold and Kamada-Kawai

An implementation of the Fruchterman-Reingold algorithm [46] is available in the *Java Universal Network/Graph Framework*[10] (JUNG) and has been integrated into the workbench using a server-side visualization agent. The visualization agent fetches the analyzed data from the SQLSpaces. Then, a copy of the original data set is converted to the graph model used by JUNG. This model is used to compute the graph drawing. Once the drawing has been completed, the visualization agent requests the coordinates of the nodes to the JUNG framework and decorates the original data set with this information. The deployment of the visualization proceeds likes with the other techniques.

An implementation of the Kamada-Kawai algorithm [67] is also available in JUNG framework and was integrated into the workbench following the same approach. Figure 3.15 shows a graph drawing produced by the Fruchterman-Reingold algorithm, while Figure 3.16 depicts a graph drawing generated by the Kamada-Kawai algorithm.

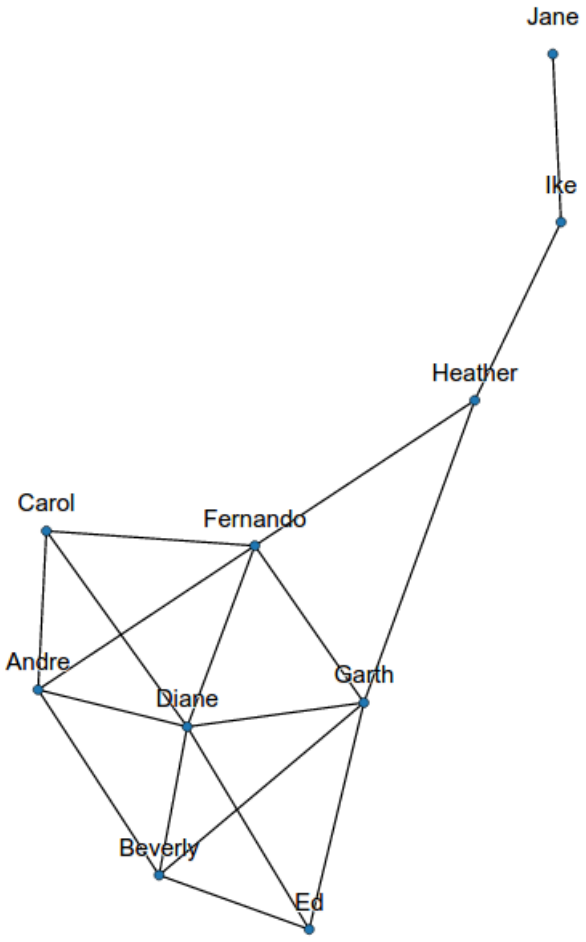

Figure 3.15.: Graph drawing created with the Fruchterman-Reingold algorithm

[10] http://jung.sourceforge.net/

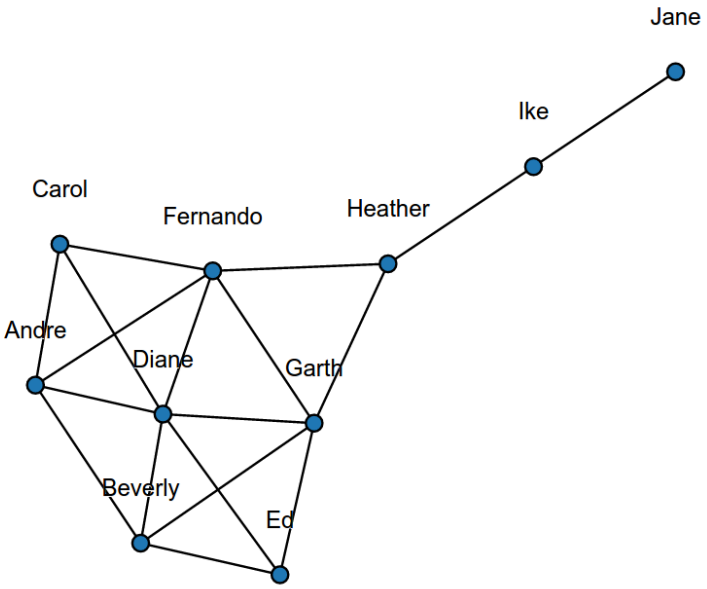

Figure 3.16.: Graph drawing created with the Kamada-Kawai algorithm

Foresighted graph layout

Another example of a technique integrated with a server-side visualization agent is the Foresighted Graph Layout [31] described in Chapter 2. On the server, the visualization agent computes the components that form the node partition and the edge partition. This information is delegated afterward to the JUNG framework in order to compute the graph drawing. Such a strategy allows the Foresighted Graph Layout to be utilized in conjunction with Circular Layout [48], Fruchterman-Reingold [46], Kamada-Kawai [67] or any other algorithm in general.

3.5. Summary

This chapter presented the framework to support flexible visualization techniques. It was developed during the run of the European Union project SISOB[11] and used to illustrate the findings of the case studies conducted in the same project. The framework addresses the technical limitations of the existing software applications to visualize dynamic networks. It allows graph drawing algorithms, layout adjustment techniques, and other visual metaphors to be integrated into the same analysis platform, i.e., the web-based analytics workbench [54]. In addition, the drawing engine can adjust its behavior using the multi-agent system of the workbench. Client-side visualization agents are suitable for simple drawings since they delegate the processing to the engine on the user interface. Server-side visualization agents are suitable for drawings that involve complex operations. The heavy computations can be performed on the server, while the engine

[11] http://sisob.lcc.uma.es/

uses the obtained results to complete the drawing on the web browser. Hybrid visualization agents are suitable for specialized visual metaphors. Furthermore, the architecture of the framework enables to deploy, update or remove visualization techniques from the web-based analytics workbench without affecting its configuration.

The workbench uses a decoration approach to handle the network data during the execution of an analysis workflow. As a result, the numeric or non-numeric attributes are added to the nodes and edges. This information is employed by the framework to adjust the graph drawing. Its filtering function removes unnecessary elements from the screen; its styling function enhances the visual appearance of the labels, nodes, and edges; while the search function permits the interactive exploration of the analyzed network. The workbench uses the discrete representation of time to create dynamic networks. Likewise, the framework follows the same approach to produce dynamic graph drawings and offers a time navigation function to explore the different periods of time. Its non-automatic mode allows the user to switch manually between consecutive periods of time, whereas the automatic mode displays the dynamic network as a movie clip. Dynamic graph drawings might change drastically at different points in time. Therefore, the framework incorporates basic layout adjustment techniques are available in the framework, however, more sophisticated approaches can be integrated through the visualization agents.

The visualization agents allow the algorithms that are available in external libraries to be integrated into the workbench. For example, the force-directed graph layout [36, 66] included in the D3.js library was integrated with a client-side visualization agent. Approaches like Fruchterman-Reingold [46] or Kamada-Kawai [67] were available in the JUNG library and integrated with a server-side visualization agent. Furthermore, the visualization agents can be used to implement algorithms that are not available in external libraries, such as the Foresighted Graph Layout [31]. The framework to support flexible visualization techniques served as a platform to study the implications of visual stability in dynamic graph drawings.

4. A mathematical model of visual stability for dynamic graph drawings

4.1. Introduction

A discrete representation of network dynamics in the form of time series of graphs (snapshots) is the approach most utilized in the analysis and visualization of dynamic social networks. During the analysis phase, the time window [60] defines the period of time in which a given network will be under observation. It is used to capture a successive sequence of states, or "snapshots", from the network timeline, while the information captured in each snapshot is analyzed through social network analysis techniques. During the visualization phase, the render window [88] reflects the information captured by the time window. For each snapshot, a graph drawing is computed separately with the algorithm of preference and the resulting sequence is presented to the user in a predefined order.

Despite the simplicity of this method, the dynamic graph drawings created from a successive sequence of snapshots have some issues. Actors, relations or patterns change their position on the canvas as the users explore the dynamic social network. This occurs because the drawing of each snapshot is different from the others in the sequence. Furthermore, actors, relations or patterns can disappear without prior information. The combination of these factors can perturb the user or even break the mental map [38].

In order to preserve the mental map, it is necessary to minimize the changes in the sequence [18]. Metrics such as the Euclidean Distance [85], the Manhattan Distance [85] or the Mental Distance [32] can be used to quantify the changes. However, the metrics only operate with the nodes in the dynamic network and ignore the edges. Moreover, they assume each node is mapped to a unique position in the Euclidean Space. Modern dynamic graph drawing algorithms, like the Foresighted Graph Layout [31], are able to map multiple nodes to the same position at different points in time. Therefore, it is not possible for the metrics to accurately quantify the changes in such a drawing.

This chapter presents a mathematical model to quantify the changes in a dynamic graph drawing. The approach takes into account mappings such as the one used in the foresighted graph layout [31] and it is inspired by the theories of visual stability described in Chapter 2. According to them [24, 30, 90, 120], an image is perceived as visually stable as long as certain "target-objects" maintain their position on the visual field. At the same time, it is important that all other elements in our sight remain constant [82]. The model uses the Euclidean Distance to verify if the nodes maintain their position in the dynamic graph drawing. In addition, it introduces other metrics to quantify the constancy of the nodes and edges during the temporal navigation.

The content of this chapter is based on the publication *"Exploring Visual Stability in Dynamic Graph Drawings: A Case Study"* [100].

4.2. Foundations of the model

The mathematical model of visual stability uses as the basis the definition of a graph [117] and a graph drawing [111], which have been described in Chapter 2. Nonetheless, it was necessary to extend them in order to cover approaches like the Foresighted Graph Layout, where multiple vertices are mapped to the same position at different points in time.

- Let f_{pv} be the mapping of the vertices V of a graph g to a **vertex logical position** in the form:

$$f_{pv} : V \rightarrow P$$

  where P is a set in the form $\{p_1, p_2, p_3, ..., p_n\}$.

- Let f_{sv} be the mapping of the vertex logical positions to a **two-dimensional Euclidean Space** in the form:

$$f_{sv} : P \rightarrow \mathbb{R}^2$$

$$f_{sv}(p) = f_{sv}(f_{pv}(v)) = (x, y)$$

- Let f_{pe}: be the mapping of the edges (u, v) of a graph g to an **edge logical position** in the form:

$$f_{pe} : E \rightarrow P \times P,$$

$$f_{pe}(e) = f_{pe}(u, v) = (f_{pv}(u), f_{pv}(v))$$

- Let f_{se}: be the mapping of the edge logical positions to a **two-dimensional Euclidean Space** in the form:

$$f_{se} : P \times P \rightarrow \mathbb{R}^2 \times \mathbb{R}^2,$$

$$\begin{aligned} f_{se}(p_u, p_v) &= (f_{sv}(p_u), f_{sv}(p_v)) \\ &= (f_{sv}(f_{pv}(u)), f_{sv}(f_{pv}(v))) \\ &= ((x_1, y_1), (x_2, y_2)) \end{aligned}$$

Therefore, a **graph drawing** is defined as a mapping of the vertices V and the edges E of a graph g to the Euclidean Space in the form:

$$d(g) = (f_{sv}(f_{pv}(V(g)), f_{se}(f_{pe}(E(g)))))$$

This extension allows the definition of a **dynamic graph drawing** [31] described in Chapter 2 to be reused without further modifications.

Model-based metrics

The mathematical model has nine metrics to quantify the visual stability of those dynamic graph drawings created with the discrete representation of time. These metrics are presented next.

Vertex set drawing active positions

The **vertex set drawing active positions** or **VDAP** calculates the percentage of **vertex logical positions** that are active in $d(g^i)$. The indicator was designed for those dynamic graph drawings using a global layout rather than techniques computing a sequence of independent drawings. The **VDAP** is defined as:

$$VDAP(g^i) = \frac{|f_{pv}(V(g^i))|}{|\cup_{i=1}^{n} f_{pv}(V(g^i))|}$$

In case the drawing technique computes a sequence of independent drawings, the **VDAP** is defined as:

$$VDAP(g^i) = \frac{|f_{pv}(V(g^i))|}{|f_{pv}(V(g^i))|} = 1$$

The values obtained for the **VDAP** are in a range between 0 and 1. Values closer to 0 indicate that a small number of vertex logical positions are active in the current drawing, while values closer to 1 indicate the opposite.

Edge set drawing active positions

The **edge set drawing active positions** or **EDAP** calculates the percentage of **edge logical positions** that are active in $d(g^i)$. The indicator was designed for those dynamic graph drawings using a global layout rather than techniques computing a sequence of independent drawings. The **EDAP** is defined as:

$$EDAP(g^i) = \frac{|f_{pe}(E(g^i))|}{|\cup_{i=1}^{n} f_{pe}(E(g^i))|}$$

In case the drawing technique computes a sequence of independent drawings, the **EDAP** is defined as:

$$EDAP(g^i) = \frac{|f_{pe}(E(g^i))|}{|f_{pe}(E(g^i))|} = 1$$

The values obtained for the **EDAP** are in a range between 0 and 1. Values closer to 0 indicate that a small number of edge logical positions are active in the current drawing, while values closer to 1 indicate the opposite.

Graph drawing offset

The **graph drawing offset** or **GDO**, calculates the changes on the coordinates of those vertex logical positions that have been mapped to the Euclidean Space.

- Let $f^i_{mv} = f_{sv}(f_{pv}(V(g^i))$ be the mapping of the vertices of $d(g^i))$ to the Euclidean space.

Thus, the **GDO** is defined as:

$$GDO(g^i, g^{i+1}) = \sum\nolimits_{v \in V(g^i) \cap V(g^{i+1})} dist(f^i_{mv}(v), f^{i+1}_{mv}(v))$$

where *dist* refers to the *euclidean distance* between two points. The values obtained for the **GDO** are higher or equal to 0. Lower values of **GDO** suggest that minimal variations on the coordinates of those vertex logical positions present in $d(g^i)$ and $d(g^{i+1})$, while higher values indicate drastic changes on the coordinates of these elements.

Vertex set stability

The **vertex set stability** or **VS** calculates the percentage of vertices from g^i that are present in g^{i+1} with the function:

$$VS(g^i, g^{i+1}) = \frac{|V(g^i) \cap V(g^{i+1})|}{|V(g^i) \cup V(g^{i+1})|}$$

The values obtained for the **VS** are in a range from 0 to 1. Values closer to 0 suggest that only a few vertices from g^i are present in g^{i+1}, while values closer to 1 indicate the opposite.

Vertex set drawing stability

The **vertex set drawing stability** or **VDS** calculates the percentage of **vertex logical positions** from $d(g^i)$ that are present in $d(g^{i+1})$ with the function:

$$VDS(g^i, g^{i+1}) = \frac{|f_{pv}(V(g^i)) \cap f_{pv}(V(g^{i+1}))|}{|f_{pv}(V(g^i)) \cup f_{pv}(V(g^{i+1}))|}$$

The values obtained for the **VDS** are in a range from 0 to 1. Values closer to 0 suggest that only a few vertex logical positions from $d(g^i)$ are present in $d(g^{i+1})$, while values closer to 1 indicate the opposite.

Edge set stability

The **edge set stability** or **ES** calculates the percentage of edges from g^i that are present in g^{i+1} with the function:

$$ES(g^i, g^{i+1}) = \frac{|E(g^i) \cap E(g^{i+1})|}{|E(g^i) \cup E(g^{i+1})|}$$

The values obtained for the **ES** are in a range from 0 to 1. Values closer to 0 suggest that only a few edges from g^i appear in g^{i+1}, while values closer to 1 indicate the opposite.

Edge set drawing stability

The **edge set drawing stability** or **EDS** calculates the percentage of **edge logical positions** from $d(g^i)$ that are present in $d(g^{i+1})$ with the function:

$$EDS(g^i, g^{i+1}) = \frac{|f_{pe}(E(g^i)) \cap f_{pe}(E(g^{i+1}))|}{|f_{pe}(E(g^i)) \cup f_{pe}(E(g^{i+1}))|}$$

The values obtained for the **EDS** are in a range from 0 to 1. Values closer to 0 suggest that only a few edge logical positions from $d(g^i)$ appear in $d(g^{i+1})$, while values closer to 1 indicate the opposite.

Vertex set degree change

The **vertex set degree change** or **VDC** calculates the variations on the degree (i.e. number of connections) of those vertices from g^i that are present in g^{i+1} with the function:

$$VDC(g^i, g^{i+1}) = \frac{\sum_{v \in V(g^i) \cap V(g^{i+1})} |\frac{C_d^i(v)}{|V(g^i)|} - \frac{C_d^{i+1}(v)}{|V(g^{i+1})|}|}{|V(g^i) \cap V(g^{i+1})|}$$

where C_d^i refers to the **degree centrality** [117] of a vertex v in g^i. The values for the **VDC** are in a range from 0 to 1. Values closer to 0 suggest that a minimal change on the degree centrality for those nodes appearing in g^i and g^{i+1}, while values closer to 1 indicate the opposite.

Vertex set drawing neighborhood change

The **vertex set drawing neighborhood change** or **VDNC** calculates the variations on the number of connections for those vertex logical positions from $d(g^i)$ that are present in $d(g^{i+1})$ with the function:

- $PV^i = f_{pv}(V(g^i))$
- $PV^{i+1} = f_{pv}(V(g^{i+1}))$
- $PE^i = f_{pe}(E(g^i))$

- $V^* = PV^i \cap PV^{i+1}$

$$VDNC(g^i, g^{i+1}) = \frac{\sum\limits_{v \in V^*} \left(| \sum\limits_{e \in PE^i} \frac{f_c(v,e)}{|PV^i|} - \sum\limits_{e' \in PE^{i+1}} \frac{f_c(v,e')}{|PV^{i+1}|} | \right)}{|V^*|}$$

and where f_c is a function in the form

$$f_c : P \times (P \times P) \rightarrow [0, 1]$$

$$f_c : (p1, (p2, p3)) = \begin{cases} 1, \text{ if } p1 = p2 \\ 1, \text{ if } p1 = p3 \\ 0, \text{ otherwise} \end{cases}$$

The values obtained for the **VDNC** are between 0 and 1. Values closer to 0 suggest that a small change on the number of connections for those vertex logical positions appearing in $d(g^i)$ and $d(g^{i+1})$, while values closer to 1 indicate the opposite.

4.3. Summary

This chapter presented a mathematical model to quantify the changes in a dynamic graph drawing. It extends the notion of mapping to cover approaches such as the foresighted graph layout [31], where multiple nodes can be assigned to the same position at different points in time. The mathematical model consists of nine metrics. Each one of them is inspired by the theories of visual stability [24, 30, 82, 90, 120]. The vertex set stability, the edge set stability along with the vertex set degree change are designed to quantify the constancy of the nodes and edges in the original dynamic network. The vertex set drawing stability, the edge set drawing stability as well as the vertex set neighborhood change are designed to quantify the constancy of the node and edge positions in the dynamic graph drawing. The vertex set drawing active positions along with the edge set drawing active position quantify the number of elements from the dynamic graph drawing that were active during the temporal navigation. Lastly, the graph drawing offset quantifies the movement of the nodes in the dynamic graph drawing during the exploration of the dynamic network. In the subsequent chapters, the mathematical model is used as a reference to explore the impact of visual stability in dynamic graph drawings.

5. Evaluation of the mathematical model of visual stability

5.1. Introduction

This chapter presents the case study to evaluate the mathematical model of visual stability described in Chapter 4. A group of 15 students was requested to track three actors in three different dynamic graph drawings. These drawings were characterized by a *constant shape*, the use of *fixed positions* and *minimal structural changes* during the temporal navigation. An eye-tracking device (SMI - RED500)[1] recorded the eye movements of the participants; a questionnaire gathered information about their experience, while the model-based metrics quantified the visual stability of the respective drawings. The data collected from the case study provided a better insight about how the visual stability of a dynamic graph drawing affects the efficiency of the visual search and the user experience.

The results obtained suggest that the visual stability of a dynamic graph drawing can affect the efficiency of the visual search and the user experience when tracking actors in a dynamic network. Dynamic graph drawings characterized by a constant shape have a low accuracy in terms of the visual search and do not provide a satisfying user experience. In contrast, dynamic graph drawings characterized by the use of fixed positions improve the accuracy of the visual search but do not provide a satisfying user experience due the constant addition and removal of elements. On the other hand, dynamic graph drawings characterized by minimal structural changes provide a satisfying user experience. However, the improvement comes at the cost of losing accuracy on the visual search.

The content of this chapter is based on the publication: *"Visual Stability in Dynamic Graph Drawings"* [102].

Basic concepts and objectives

In recent years, researchers have incorporated the use of eye-tracking devices into usability studies [15, 63, 93, 97]. These devices record the eye movements performed by the users as they complete a series of tasks, which are often focused on locating objects on the screen. The data collected from the eye-tracking device allows the researchers to have a better insight about the user's attention [84] and also to evaluate the efficiency of the visual search [95].

The visual search refers to the eye movements a user performs to locate a desired object [56] and its efficiency is evaluated using the attributes of the same movements. A fixation

[1] http://www.smivision.com

is when the eyes are essentially stationary [64]. A saccade is a rapid eye movement from one fixation to another [64], whereas a scan path describes a sequence of alternating fixations and saccades [64]. According to Goldberg [56], the occurrence and duration of the attributes aforementioned determine the efficiency of the visual search. Through case studies [26, 56, 57, 94, 95], it has been found that visual searches with lower occurrences and durations are equal to an efficient visual search. This is because the user is finding the desired object in a short period of time with the minimal number of eye movements. In contrast, visual searches with higher occurrences and durations imply the opposite.

The user experience is other of the aspects evaluated in usability studies. It takes into account the thoughts, feelings, and the perception that result from the user's interaction with a system [3]. The user experience has three main characteristics [3]:

- It involves a user.
- The user interacts, with a product, system or anything with an interface.
- The user's experience is of interest, and observable or measurable.

The metrics to evaluate the user experience focus on specific criteria such as task success, errors, response time and user satisfaction [3]. However, in order to use them, it is necessary to have the appropriate data. Self-reported data provides the most important information about the user's perception of a system and the interaction with it. It allows to understand how the user "feels" about the system. In many situations, this represents a priority regarding the user experience. According to Albert [3], the users can spend a considerable amount of time performing tasks related to the system. Nevertheless, if the experience makes them happy that is the only thing that matters.

Self-reported data can be collected using rating scales [3]. For example, the Likert scale [77] presents five statements. Each one of them is related to a level of agreement. The value of 1 represents a strong disagreement, 2 a disagreement, 3 neither an agreement nor a disagreement, 4 an agreement and 5 a strong agreement. The Likert scale is frequently used in questionnaires, where each question evaluates a characteristic of the system. The questionnaire is given to the user once he has finished interacting with the system and the data is analyzed using statistical techniques. As a result, a better insight of the user experience is obtained. The following case study aims to evaluate the mathematical model of visual stability described in Chapter 4. The main objective was to determine how such a property affects the efficiency of the visual search and the user experience in terms of the user's satisfaction.

Hypothesis

The research hypothesis for the case study implied that *visually stable dynamic graph drawings improve the efficiency of the visual search and the user experience when tracking actors in a dynamic network.*

Table 5.1.: Details of the dynamic network used in the case study

	Snapshots				
	1	**2**	**3**	**4**	**5**
Number of nodes	16	26	27	5	6
Number of edges	51	86	97	13	16

5.2. Application of the model-based metrics to the data set

Asterisk[2] is an open source framework for creating communications applications. It was created by Mark Spencer back in 1999. Asterisk not only allows the telephones to communicate with each other but also to connect to other popular services, such as the public switched telephone network (PSTN) and Voice over Internet Protocol (VoIP). Furthermore, there is a huge community of developers which are constantly improving the framework [126].

The dynamic network used in this case study was extracted from the developer mailing list of the Asterisk framework. The developers were considered as the nodes and the e-mail exchange between them as the edges. It was necessary to reduce the number of these elements due to the size of the data set. The dynamic network was uploaded into the web-based analytics workbench [54] presented in Chapter 3 and a K-core [108] with a value of 4 was used in the reduction process. Five snapshots were selected from the resulting network. Each one of them represented a month of communications between the developers. The details are shown in Table 5.1.

After the reduction process, the dynamic network was uploaded into the framework to support flexible visualization techniques presented in Chapter 3. Three dynamic graph drawings were created. Each one of them had a characteristic related to the theories of visual stability described in Chapter 2. A *Circular Layout* (CL) is selected because it provides a dynamic graph drawing with a *constant shape*. However, it changes the positions of the nodes during the exploration of the dynamic network. A *Circular Super Graph* (CSG) is selected because it assigns a *fixed position* in the Euclidean Space to every actor in the dynamic network. Nonetheless, it does not possesses a constant shape. Lastly, a *Foresighted Circular Layout* (FCL) is chosen because it provides a dynamic graph drawing with the *minimal structural changes* during the exploration of the dynamic network. Nevertheless, it allows multiple actors to occupy the same position at different points in time. It was decided to employ circular drawings, to evaluate the same shape under distinct conditions. Figures 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 5.10, 5.11, 5.12, 5.13, 5.14 and 5.15 illustrate the drawings used in the case study.

The model-based metrics were computed for the three dynamic graph drawings. They were calculated as it is described in Chapter 4. In addition, the average value of every metric was computed to obtain the overall visual stability of the dynamic graph drawings used in the case study.

[2] www.asterisk.org

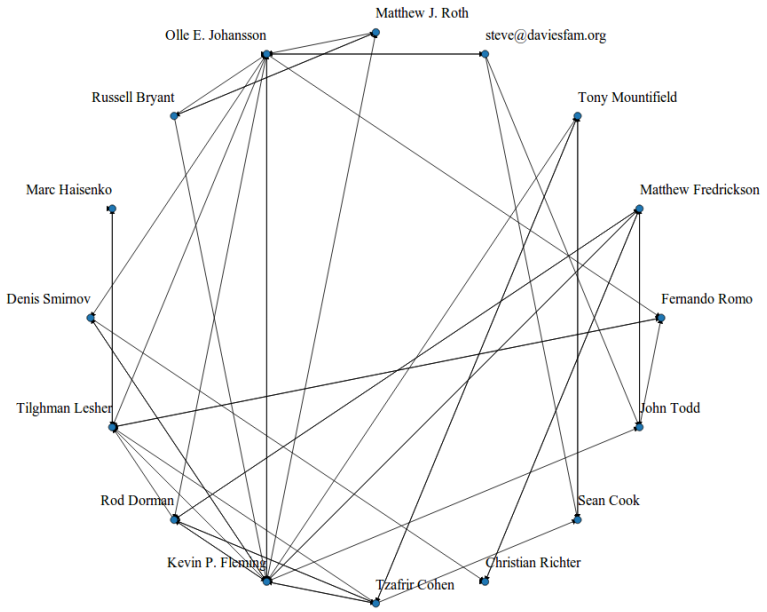

Figure 5.1.: Dynamic graph drawing produced by the circular layout for snapshot 1

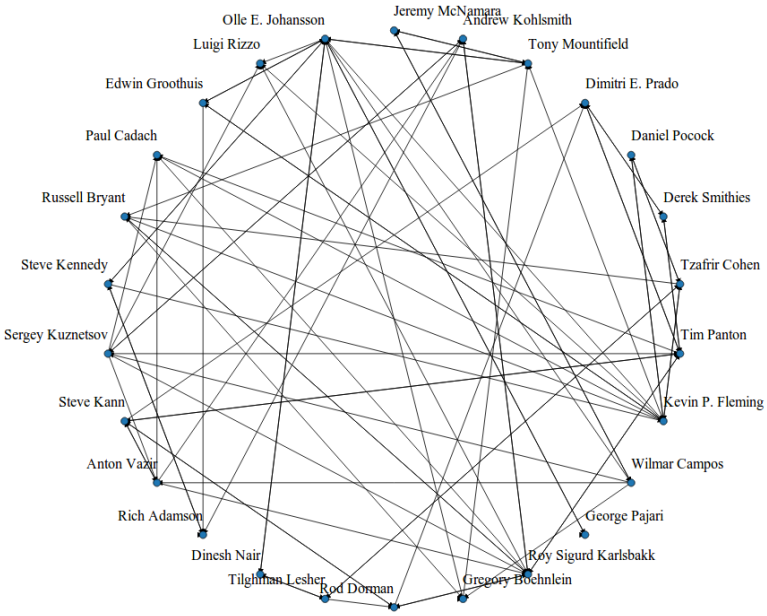

Figure 5.2.: Dynamic graph drawing produced by the circular layout for snapshot 2

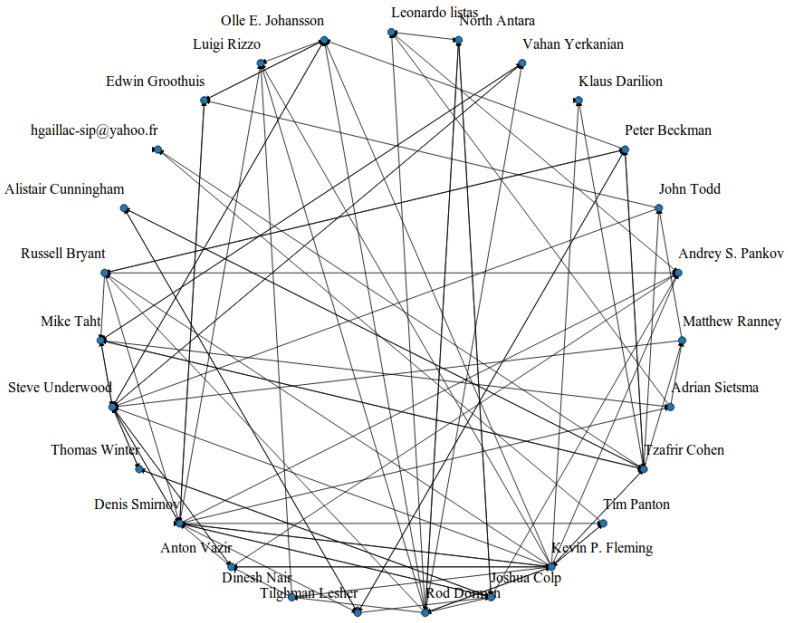

Figure 5.3.: Dynamic graph drawing produced by the circular layout for snapshot 3

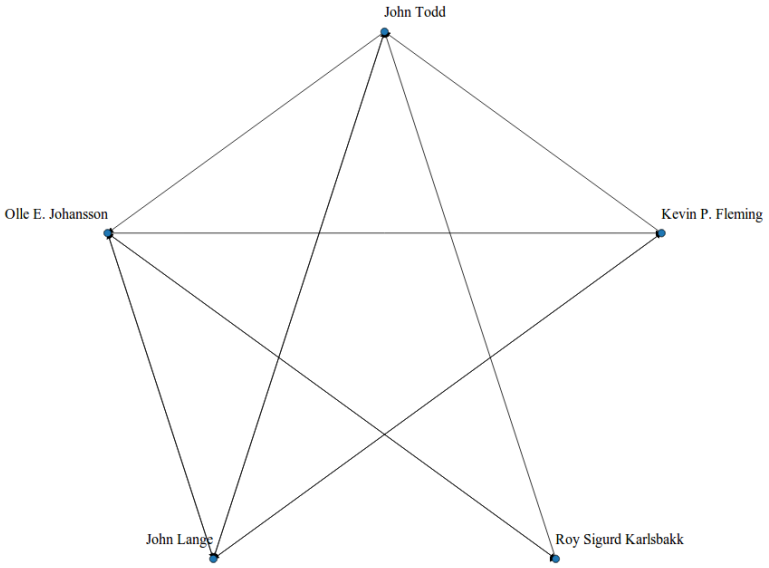

Figure 5.4.: Dynamic graph drawing produced by the circular layout for snapshot 4

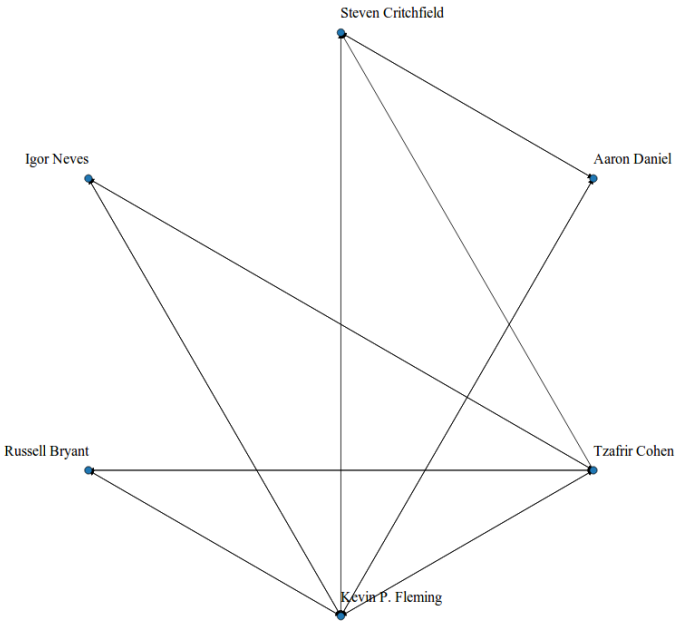

Figure 5.5.: Dynamic graph drawing produced by the circular layout for snapshot 5

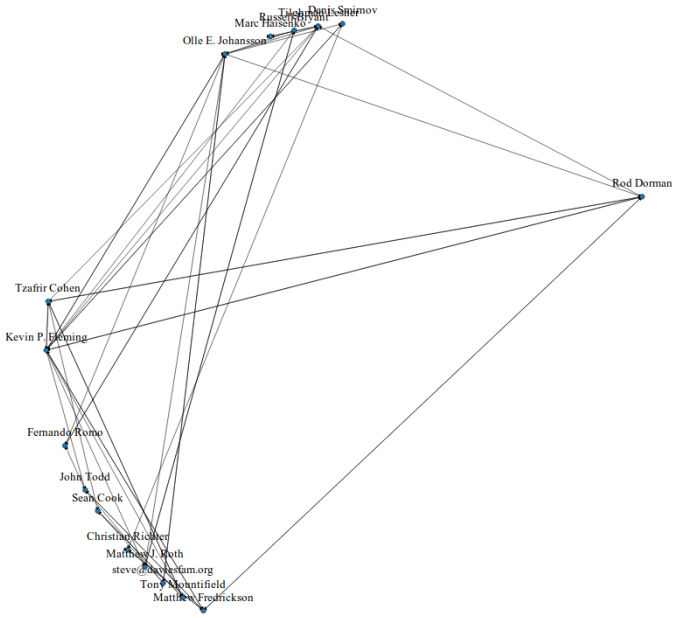

Figure 5.6.: Dynamic graph drawing produced by the circular super graph for snapshot 1

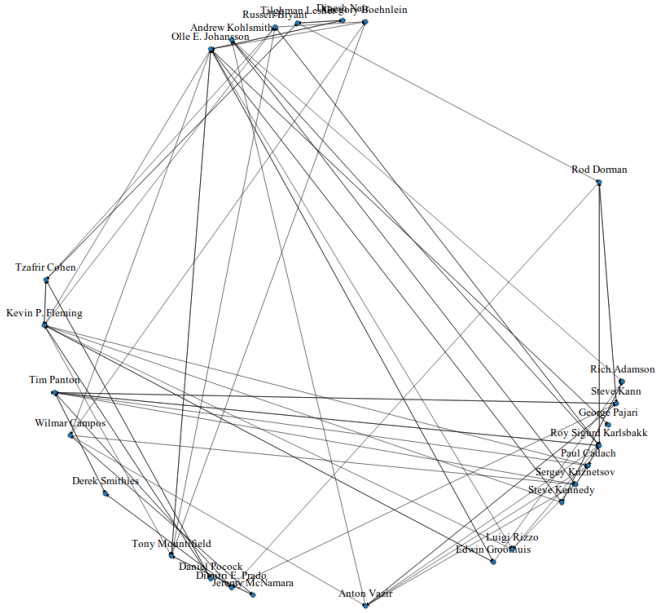

Figure 5.7.: Dynamic graph drawing produced by the circular super graph for snapshot 2

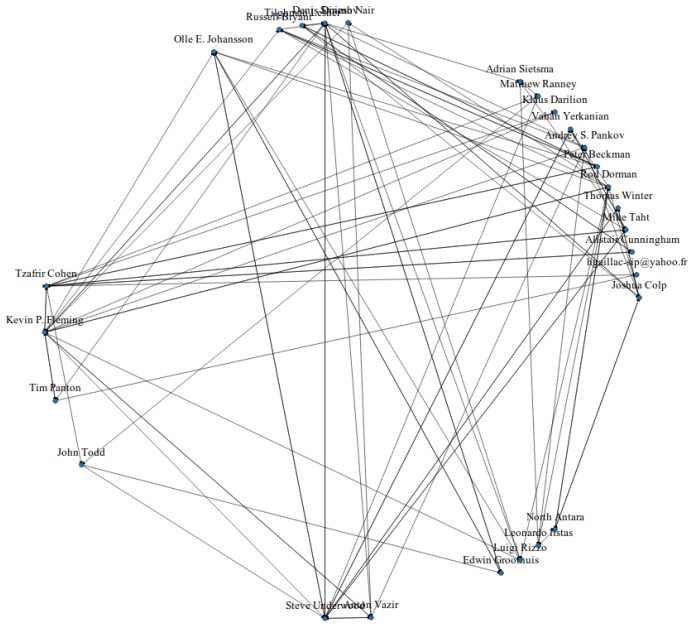

Figure 5.8.: Dynamic graph drawing produced by the circular super graph for snapshot 3

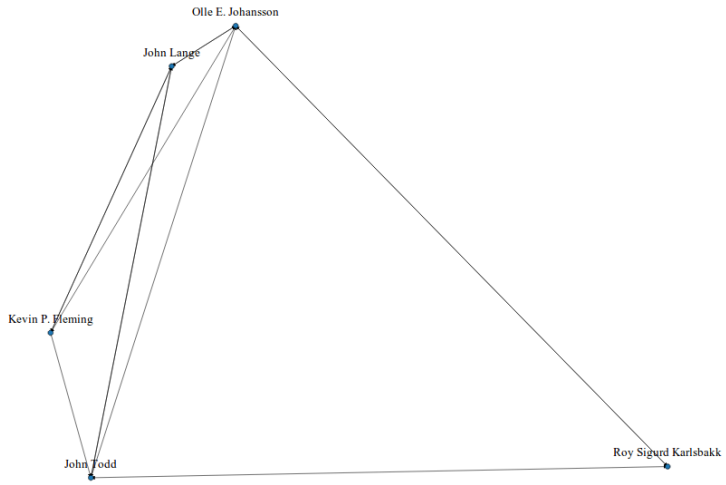

Figure 5.9.: Dynamic graph drawing produced by the circular super graph for snapshot 4

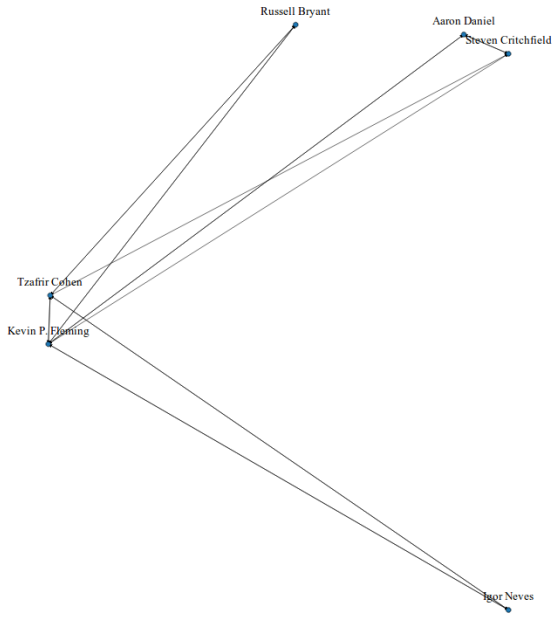

Figure 5.10.: Dynamic graph drawing produced by the circular super graph for snapshot 5

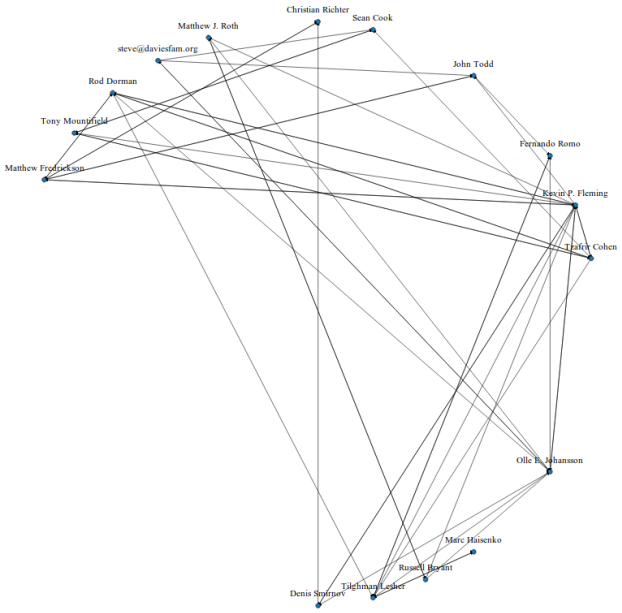

Figure 5.11.: Dynamic graph drawing produced by the foresighted circular layout for snapshot 1

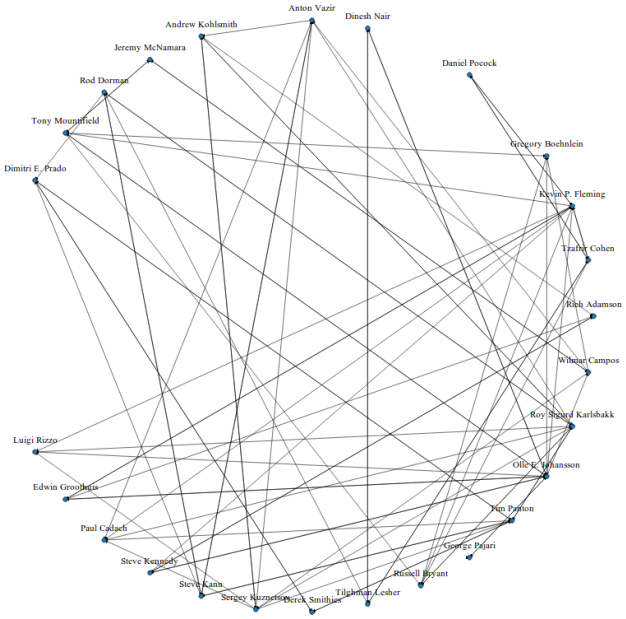

Figure 5.12.: Dynamic graph drawing produced by the foresighted circular layout for snapshot 2

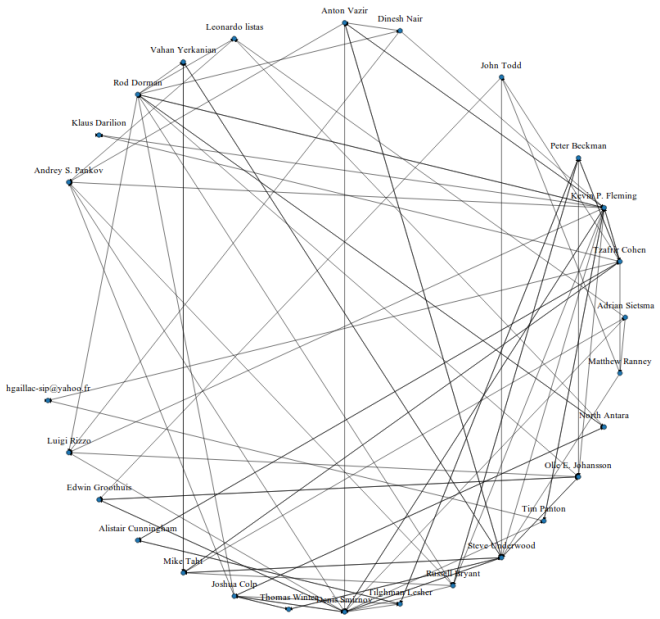

Figure 5.13.: Dynamic graph drawing produced by the foresighted circular layout for snapshot 3

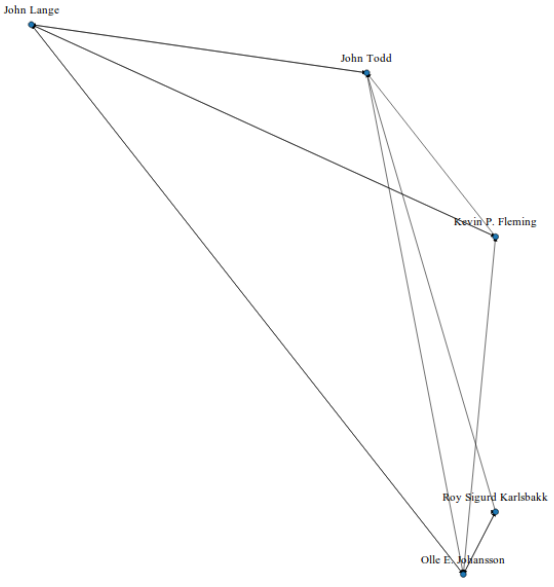

Figure 5.14.: Dynamic graph drawing produced by the foresighted circular layout for snapshot 4

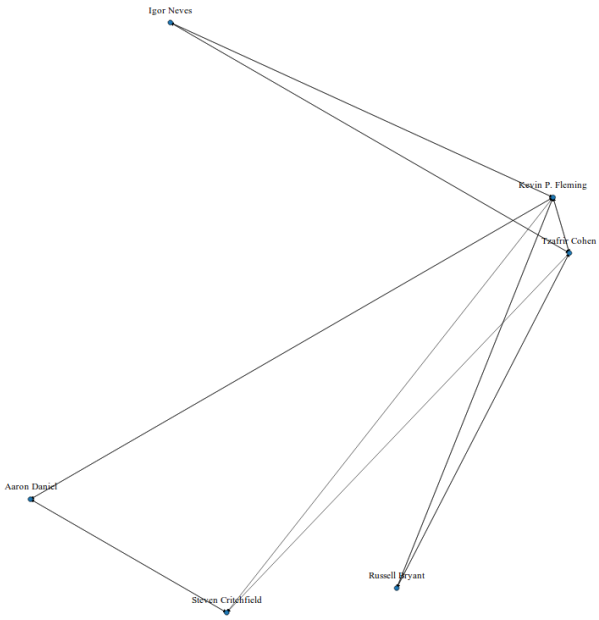

Figure 5.15.: Dynamic graph drawing produced by the foresighted circular layout for snapshot 5

5.3. Eye-tracking metrics

The data collected by the eye-tracking device was analyzed with a series of metrics, which have been proven to measure cognitive load and efficiency of the visual search in search tasks [26, 56, 57, 94, 95]. Two distinct settings were used in the analysis. The first one applied nine metrics to the overall drawing area. These metrics are listed below:

- *Number of fixations* - It refers to the number of times the eyes are essentially stationary [64]. A larger number of fixations indicate that the user sampled other objects prior to the desired target [56]. Thus, more fixations indicate a less efficient visual search [94].
- *Duration of fixations* - It refers to the time a user spends interpreting or relating a component on the screen to its internalized representation [56]. A longer duration suggests difficulties in the interpretation process or in the extraction of information [94].
- *Number of saccades* - It refers to the number of rapid eye movement from one fixation to another [64]. A large number of saccades suggest a more extensive and inefficient visual search [56].
- *Duration of saccades* - It refers to the time spent for the eyes to move from one fixation to another [15]. A longer duration suggests a more extensive and inefficient visual search [56].
- *Saccadic amplitude* - It refers to the distance between two fixation points [56]. A larger amplitude indicates more meaningful cues, as attention is drawing from a distance [96].

- *Scan path length* - It refers to the distance covered by a sequence of alternating fixations and saccades [64]. A longer distance suggests a more extensive and inefficient visual search [56, 96].
- *Scan path duration* - It refers to the processing time during the visual search, since more time is spent in the fixation than in the saccades [56]. A longer scan path duration suggests an inefficient visual search [96].
- *Spatial density* - It refers to the coverage of an image or user interface determined by the spatial distribution of the gaze points samples [56]. Evenly spread samples through the display indicate an extensive visual search with an inefficient scan path, whereas targeted samples in a small area reflect a direct and efficient search [56].
- *Fixations/Saccade ratio* - It refers to a content independent ratio, which compares the time spent processing a component (fixations) to the time spent searching for the desired component (saccades) [56]. A higher ratio indicates that there was either more processing or less search activity [56].

In the context of eye-tracking studies, the areas of interest or AOIs are parts of a stimulus that contain objects of potential interest to the researcher [15, 57]. The second setting applied two metrics to the areas where the requested actors were located. This AOIs did not maintain a fixed position for the scenarios subject to evaluation. Instead, they were placed in the Euclidean Space according to the rules of the dynamic graph drawings. The metrics applied to the AOIs are listed below:

- *Time to first fixation* - It refers to the time spent to fixate a specific object or desired area [96]. A larger time to the first fixation means that an object does not have attention-getting properties [96].
- *Fixations on target* - It refers to a ratio that measures the accuracy of the visual search. The ratio is calculated counting the number of fixations falling on a designated target and then, dividing by the number of fixations in the overall drawing area. A lower ratio indicates low accuracy in the visual search [56].

The metrics applied to the overall drawing area and the AOIs were compared for the three drawings. A T-test was used to look for statistical significance and also to determine which dynamic graph drawing has the best efficiency in terms of the visual search according to the metrics.

5.4. Questionnaires

The students were requested to answer a questionnaire about their experience with the dynamic graph drawings used in the respective scenario. Each question was rated with a Likert scale [77] of five points. The value of 1 represented a strong disagreement, the value of 3 neither disagreement nor agreement and the value of 5 a strong agreement. The questionnaire was designed to evaluate a specific set of criteria from the dynamic graph drawings, which are listed below:

Criteria subject to evaluation

- C1 - The drawing displayed the specific actors in the same position (corresponds to the question number 1).
- C2 - The drawing technique was useful to track actors in a dynamic network (corresponds to the question number 4).
- C3 - The changes in the drawing were noticeable (corresponds to the question number 5).
- C4 - It was easy to locate an actor on the drawing (corresponds to the question number 3).
- C5 - The addition or removal of actors and relations from the drawing was distracting (corresponds to the question number 6).

Despite having a total of six questions, it was decided to ignore the question number two from the questionnaire. Such a decision was due to the information provided by the question number three, which was more useful for the study. In addition, the questionnaire included a series of open questions where the users could report positive and negative aspects of the dynamic graph drawings. This information was analyzed with a card sorting technique [3] in order to build categories that describe more adequately the user experience.

5.5. Study setup

The case study has three scenarios. Each one of them illustrates the reduced dynamic network from developer mailing list of the Asterisk framework with one of the dynamic graph drawings described beforehand. *Scenario 1* use the *Circular Layout* (CL), *Scenario 2* use the *Circular Super Graph* (CSG), while *Scenario 3* use the *Foresighted Circular Layout* (FCL). The order in which the scenarios were presented to the participants of the case study was assigned randomly.

The participants of the case study were a group of 15 students aged between 22 and 28. They were from the fields of computer science and applied cognitive science. All students had a basic knowledge about social network analysis techniques but were not familiar with the theories of visual stability nor the mental map.

The students had to complete 13 tasks for each scenario. These tasks requested to track three developers of the Asterisk framework: *Tony Mountifield* (Actor A), *Kevin P. Fleming* (Actor B) and *Rod Dorman* (Actor C). The developers were selected not only because they belonged to the core of the community, but also because they disappeared from the dynamic network at different points in time.

The first task requested the students to locate the three developers in the dynamic graph drawing. Task 2 to 4 were applied after a transition to the second snapshot. Tasks 5 to 7 were applied after a transition to the third snapshot. Tasks 8 to 10 were applied after a transition to the fourth snapshot. Finally, tasks 10 to 13 were applied after a transition to the fifth snapshot. All the tasks requested relocate the three developers in the dynamic graph drawing. Furthermore, they asked if they appeared in the same

position as in the previous snapshots and if the changes in the dynamic graph drawing were distracting.

There was no time limit for completing the tasks or the scenarios. Yet, the average duration per student for the three scenarios was 30 minutes. The tasks along with the questionnaire used in the case study are available in Appendix A.

5.6. Analysis and results

Model-based metrics

As illustrated in Table 5.2, the *CSG* presented the lowest number of active positions. Only 19% (0.198) of the vertex positions were active (VDAP), while 11% (0.112) of the edge positions (EDAP) were in the same state. In contrast, the *FCL* presented 48% (0.485) of the vertex positions as active, while 14% (0.143) of the edge positions were in the same state. The *CL* relies on a successive sequence of graph drawings, which are independent of each other. As a result, the dynamic graph drawing had 100% (1.000) of the vertex positions as active (VDAP) and 100% (1.000) edge logical positions (EDAP) in the same state.

Another difference between the dynamic graph drawings was related to the movement of the nodes in the Euclidean Space. According to the model of visual stability, the *FCL*, and the *CSG* had a graph drawing offset (GDO) of 0.000. The reason for this is that both assign the nodes to a fixed location in the Euclidean Space. In contrast, the *CL* assigns the nodes to a different location in each snapshot of the dynamic network. As a consequence, it presented a graph drawing offset (GDO) of 2564.683. This suggests that the nodes in the *CL* are in constant movement.

Dynamic graph drawing algorithms map respectively an actor and a relation to a unique node and edge position in the Euclidean Space. This characteristic is found in the *CL* and in the *CSG*. According to the model of visual stability, both approaches had a vertex set stability (VS) of 16% (0.165), an edge set stability (ES) of 1% (0.015), while the vertex set degree change (VDC) was of 41% (0.413). The same results were found for the vertex set drawing stability (VDS), the edge set drawing stability (EDS) and the vertex set drawing neighborhood change (VDNC) due to their mapping.

In contrast to the previous techniques, the *FCL* maps multiple actors and relations to the same node and edge positions at different points in time. It presented a vertex set stability (VS) of 16% (0.165), an edge set stability (ES) of 1% (0.015) and a vertex set degree change (VDC) was of 41% (0.413). However, the vertex set drawing stability (VDS) was of 49% (0.493), the edge set drawing stability (EDS) was of 4% (0.043) and the vertex set drawing neighborhood change (VDNC) was of 0.40% (0.403). This suggests that the *FCL* maintains more elements on the screen despite the network is actually changing.

Table 5.2.: Average visual stability calculated for the three drawings under study

Dimensions of the model	CL	CSG	FCL
VDAP vertex set drawing active positions	1.000	0.198	0.485
EDAP edge set drawing active positions	1.000	0.112	0.143
GDO graph drawing offset	2564.683	0.000	0.000
VS vertex set stability	0.165	0.165	0.165
VDS vertex set drawing stability	0.165	0.165	0.493
ES edge set stability	0.015	0.015	0.015
EDS edge set drawing stability	0.015	0.015	0.043
VDC vertex set degree change	0.413	0.413	0.413
VDNC vertex set drawing neighborhood change	0.413	0.413	0.403

Questionnaires

The data collected from the questionnaires provided an insight about the user experience in the different scenarios. It was computed the average value of the user ratings per criterion and they lied within a range of [1, 5] with a mean value of 3. All the criteria are perceived as positive, which means that a rating of 5 is perceived as the best, while a rating of 1 is perceived as the worst.

Scenario 1 presented the dynamic social network with the *CL*, obtaining *09:14* minutes as the average execution time for all the tasks. As illustrated in Figure 5.16, the *CL* was considered not to be useful for tracking actors in a dynamic social network (C2 = 2.667). The students did not perceive that the actors were assigned to a fixed position in the Euclidean Space (C1 = 2.467) and had problems to locate the requested actors (C4 = 2.733). The changes in the dynamic graph drawing were noticeable (C3 = 3.267), but the addition and removal of actors or relations were not distracting (C5 = 2.533).

Considering the information from the open questions, six categories were found for the *CL* using the card sorting technique. The category called *drawing characteristics* reported that the drawing was not showing the changes in the dynamic network; an aspect which was considered as negative by the users. The category called *readability* reported that the names of the actors in the drawing were easy to read. This aspect was considered as positive by the users. The category called *location* reported that it was hard to locate an actor during the temporal navigation even though the nodes had a good separation distance between each other. These aspects were considered as positive and negative respectively. The category called *complexity* reported that the drawing was confusing and that more concentration was required to locate an actor. Both aspects were

considered as negative by the users. The category called *dynamics* reported that the drawing changed the positions of the nodes during the temporal navigation; an aspect which was considered as negative by the users. Lastly, the category *drawing overview* reported that the drawing was not complex and offered a good overview of the period of time under exploration. Both aspects were considered as positive.

Scenario 2 presented the dynamic network with the *CSG*, obtaining *09:11* minutes as the average execution time for all the tasks. As illustrated in Figure 5.16, the *CSG* was considered not to be useful for tracking actors in a dynamic social networks (C2 = 2.867). Despite this fact, the students positively received that the actors were assigned to a fixed position in the Euclidean Space (C1 = 3.400) and mentioned that it was easier to locate the requested actors (C4 = 3.267). The changes in the dynamic graph drawing were noticeable (C3 = 3.067), while the addition and removal of actors or relations were a bit more distracting (C5 = 2.667).

Considering the information from the open questions, six categories were found for the *CSG*. The category called *drawing characteristics* reported that the drawing showed the changes in the dynamic network; an aspect which was considered as negative by the users. The category called *readability* reported that the names of the actors were hard to read. This aspect was considered as negative by the users. The category *location* reported that the nodes did not have a good separation distance between each other, still, it was easy to locate an actor. These aspects were considered as negative and positive respectively. The category called *complexity* reported that the drawing presented too many nodes and also too many edges. This aspect was considered as negative by the users. The category called *drawing overview* reported that the drawing was complex and the users considered it as negative. Lastly, the category called *dynamics* reported that the actors always maintained a fixed position in the Euclidean Space. This was considered as a positive aspect.

Scenario 3 presented the dynamic network with the *FCL*, obtaining *09:09* minutes as the average execution time for all the tasks. As it is shown in Figure 5.16, the *FCL* was considered to be the most useful technique for tracking actors in a dynamic social network (C2 = 3.267). The students perceived that the actors were assigned to a fixed position in the Euclidean Space (C1 = 3.867) and it was easier to locate the requested actors in comparison to the *CSG* and the *CL* (C4 = 3.667). Furthermore, the changes in the drawing were not noticeable (C3 = 2.400), while the addition and removal of actors or relations were found not to be distracting (C5 = 2.333).

In regard to the open questions, six categories were found for the *FCL*. The category called *dynamics* reported that it was, in fact, confusing to observe multiple actors occupy the same position at different points in time, still, it was perceived that the drawing maintained the positions of the nodes in the Euclidean Space. These aspects were considered to be negative and positive, respectively. The category called *readability* reported that the names of the actors were easy to read; an aspect which was considered as positive by the users. The category *drawing overview* reported that the drawing was not complex and that it provided a good overview of the period of time under exploration. These aspects were considered as positive. The category *location* reported that the nodes had a good separation distance between each other. Furthermore, it was easy to locate the desired actor. Both factors were considered as positive. The category *complexity* reported that the technique was confusing and this aspect was

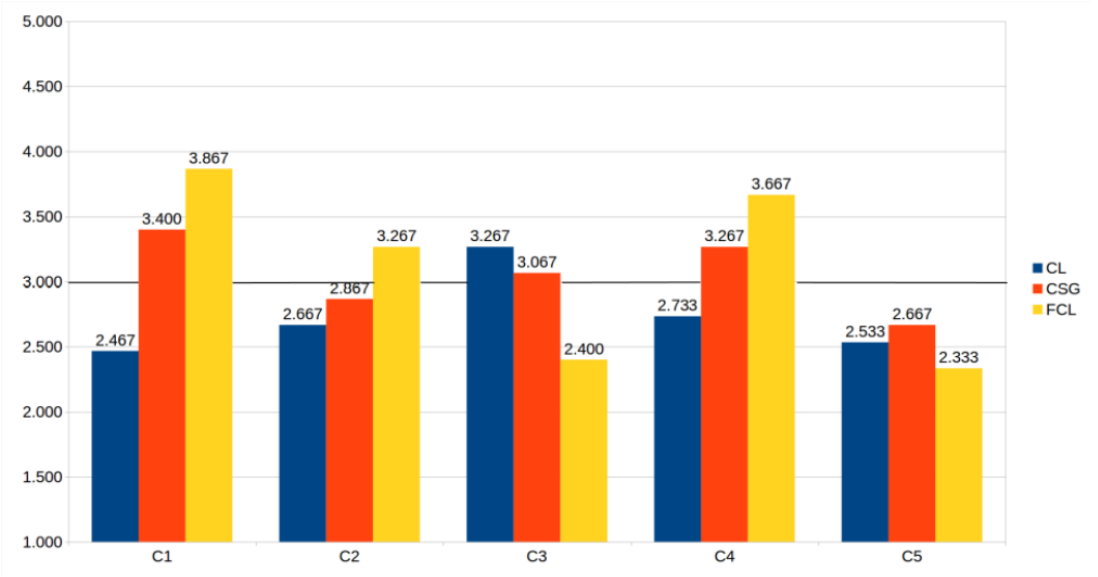

Figure 5.16.: Results of the questionnaires for the five criteria

considered to be negative. Lastly, the category called *drawing characteristics* reported that the drawing did not show the changes in the dynamic network and was considered as negative.

Eye-tracking metrics on the overall drawing area

Number of fixations

As illustrated in Table 5.3, the *CSG* obtained the best performance in terms of the number of fixations. It had an average of 64.080 fixations, followed by the *FCL* with 70.453 and the *CL* with 81.787. This result suggests that the students performed fewer fixations using the *CSG* in comparison to the other approaches. However, the difference was found not to be statistically significant as shown in Table 5.4.

Duration of fixations

As illustrated in Table 5.5, the *CL* obtained the best performance in terms of the duration of the fixations. It had an average time of 0.234 seconds per fixation, followed by the *FCL* with 0.267 seconds and the *CSG* with 0.289 seconds. This result suggests that using the *CL* the students processed faster the elements in the drawing. Furthermore, the difference was found to be statistically significant as shown in Table 5.6. The *CL* offered a better processing time in comparison to the *CSG* ($[Z = -3.010b, p = 0.003]$) and the *FCL* ($[Z = -3.408b, p = 0.001]$). Similarly, the *FCL* offered a better processing time in comparison to the *CSG* ($[Z = -2.158c, p = 0.031]$).

Table 5.3.: Number of fixations on the overall drawing area

Number of Fixations			
ID	**CL**	**CSG**	**FCL**
0584134975	130.400	60.800	88.800
0761924407	56.800	41.600	71.000
0935711842	61.200	91.200	81.000
1228890554	102.000	54.200	55.800
1325908148	131.200	102.000	88.000
1921419896	63.600	55.200	64.400
2415213624	86.000	56.400	75.600
318108245	65.000	74.000	60.200
3723364039	61.600	43.600	58.400
4430417015	88.000	100.800	111.000
6477520522	88.000	54.800	60.000
8593499818	52.200	30.800	53.200
9626467901	57.600	51.000	65.400
9754525723	113.200	86.400	71.600
9849337422	70.000	58.400	52.400
Average	**81.787**	**64.080**	**70.453**
Standard Deviation	**26.676**	**21.854**	**16.231**

Table 5.4.: T-test for the number of fixations on the overall drawing area

	T-test		
	CL-CSG	**CL-FCL**	**CSG-FCL**
Z	-2.272b	-1.477b	-1.477c
p	0.023	0.140	0.140

Table 5.5.: Duration of fixations in seconds on the overall drawing area

Duration of Fixations			
ID	**CL**	**CSG**	**FCL**
0584134975	0.289	0.352	0.310
0761924407	0.236	0.319	0.303
0935711842	0.170	0.172	0.208
1228890554	0.246	0.295	0.290
1325908148	0.264	0.345	0.292
1921419896	0.262	0.302	0.288
2415213624	0.210	0.287	0.222
318108245	0.283	0.371	0.324
3723364039	0.191	0.263	0.203
4430417015	0.224	0.264	0.243
6477520522	0.177	0.209	0.179
8593499818	0.241	0.183	0.320
9626467901	0.239	0.326	0.299
9754525723	0.218	0.265	0.221
9849337422	0.257	0.388	0.298
Average	**0.234**	**0.289**	**0.267**
Standard Deviation	**0.036**	**0.065**	**0.048**

Table 5.6.: T-test for the duration of fixations on the overall drawing area

	T-test		
	CL-CSG	**CL-FCL**	**CSG-FCL**
Z	-3.010b	-3.408b	-2.158c
p	0.003	0.001	0.031

Table 5.7.: Number of saccades on the overall drawing area

Number of Saccades			
ID	**CL**	**CSG**	**FCL**
0584134975	130.200	59.800	87.600
0761924407	56.000	40.000	70.200
0935711842	80.200	95.400	93.600
1228890554	98.800	53.800	56.000
1325908148	130.000	100.200	87.200
1921419896	63.800	55.400	65.200
2415213624	85.400	55.000	76.000
318108245	62.600	73.600	60.600
3723364039	63.800	44.600	63.200
4430417015	82.200	97.800	106.600
6477520522	88.200	54.400	61.000
8593499818	52.600	32.400	52.600
9626467901	60.000	54.000	63.200
9754525723	113.600	92.000	72.600
9849337422	68.800	56.800	50.400
Average	**82.413**	**64.347**	**71.067**
Standard Deviation	**25.629**	**22.014**	**16.247**

Table 5.8.: T-test for the number of saccades on the overall drawing area

	T-test		
	CL-CSG	**CL-FCL**	**CSG-FCL**
Z	-2.613b	-1.538b	-1.591c
p	0.009	0.124	0.112

Number of saccades

As illustrated in Table 5.7, the *CSG* obtained the best performance in terms of the number of saccades. It had an average of 64.347 saccades, followed by the *FCL* with 71.067 and the *CL* with 82.413. This result indicates that the students performed less eye movements using the *CSG*. Moreover, the difference was found to be statistically significant as depicted in Table 5.8. The *CSG* presented less eye movements in comparison to the *CL* ($[Z = -2.613b, p = 0.009]$). However, there was no evidence that the *CL* was superior to the *FCL*. Likewise, there was no evidence that the *FCL* was superior to the *CSG*.

Duration of saccades

As illustrated in Table 5.9, the *FCL* obtained the best performance in terms of the duration of the saccades. It had an average time of 0.036 seconds per saccade, followed by the *CL* with 0.039 and the *CSG* with 0.041. This result suggests that the students performed faster eye movements using the *FCL*. Nonetheless, the difference was found not to be statistically significant as shown in Table 5.10.

Table 5.9.: Duration of saccades in seconds on the overall drawing area

Duration of Saccades			
ID	**CL**	**CSG**	**FCL**
0584134975	0.027	0.029	0.021
0761924407	0.024	0.031	0.031
0935711842	0.090	0.103	0.072
1228890554	0.045	0.054	0.038
1325908148	0.031	0.023	0.035
1921419896	0.031	0.029	0.023
2415213624	0.028	0.027	0.035
318108245	0.028	0.024	0.023
3723364039	0.057	0.072	0.067
4430417015	0.031	0.037	0.035
6477520522	0.038	0.037	0.032
8593499818	0.048	0.052	0.036
9626467901	0.036	0.031	0.023
9754525723	0.040	0.030	0.049
9849337422	0.024	0.033	0.019
Average	**0.039**	**0.041**	**0.036**
Standard Deviation	**0.017**	**0.022**	**0.016**

Table 5.10.: T-test for the duration of saccades on the overall drawing area

	T-test		
	CL-CSG	**CL-FCL**	**CSG-FCL**
Z	-1.022b	-1.079c	-1.647c
p	0.307	0.281	0.112

Table 5.11.: Saccadic amplitude in pixels on the overall drawing area

Saccadic Amplitude			
ID	**CL**	**CSG**	**FCL**
0584134975	128.273	122.679	115.554
0761924407	152.025	153.383	139.168
0935711842	128.078	134.091	117.066
1228890554	159.100	175.732	112.089
1325908148	116.465	111.324	130.782
1921419896	142.803	150.007	144.767
2415213624	163.785	156.578	136.304
318108245	129.628	123.632	125.218
3723364039	162.351	162.582	114.495
4430417015	138.036	139.104	101.296
6477520522	126.003	137.615	121.733
8593499818	131.889	114.431	122.401
9626467901	164.892	126.269	108.090
9754525723	138.590	118.543	119.751
9849337422	123.022	160.454	116.627
Average	**140.329**	**139.095**	**121.689**
Standard Deviation	**16.251**	**19.787**	**11.894**

Table 5.12.: T-test for the saccadic amplitude on the overall drawing area

	T-test		
	CL-CSG	**CL-FCL**	**CSG-FCL**
Z	-0.170b	-2.840b	-2.385b
p	0.865	0.005	0.017

Saccadic amplitude

The *FCL* obtained the best performance in terms of the saccadic amplitude, as depicted in Table 5.11. It had an average distance of 121.689 pixels, followed by the *CSG* with 139.095 and the *CL* with 140.329. This result suggests that the saccades performed by the students covered shorter distances using the *FCL*. Furthermore, the difference was found to be statistically significant as displayed in Table 5.12. The *FCL* presented shorter saccadic aimplitudes in comparison to the *CL* ($[Z = -2.840b, p = 0.005]$) and the *CSG* ($[Z = -2.385b, p = 0.017]$). However, there was no evidence indicating that the *CL* was superior to the *CSG* or vice versa.

Scan path length

Table 5.13 illustrates that the *CSG* obtained the best performance in terms of the scan path length. It had an average distance of 3041.666 pixels, followed by the *FCL* with 3173.702 and the *CL* with 3441.962. This result suggests that the *CSG* provided the shortest distance during the visual search. Nonetheless, the difference was found not to be statistically significant as shown in Table 5.14.

Table 5.13.: Scan path length in pixels on the overall drawing area

Scan Path Length			
ID	**CL**	**CSG**	**FCL**
0584134975	3862.484	3647.529	4736.939
0761924407	4138.323	3115.654	4103.755
0935711842	902.310	559.925	548.678
1228890554	3028.258	1712.152	2299.914
1325908148	5069.930	2904.104	2374.845
1921419896	3028.840	2400.756	2900.893
2415213624	7563.217	5022.070	5634.977
318108245	2160.580	3914.229	1884.659
3723364039	2673.858	1725.844	1995.837
4430417015	2578.134	3556.193	1887.240
6477520522	3052.067	2415.814	3049.216
8593499818	2829.828	2020.186	3196.505
9626467901	2175.133	2755.279	3891.857
9754525723	4338.990	4252.185	5150.365
9849337422	4227.483	5623.063	3949.852
Average	**3441.962**	**3041.666**	**3173.702**
Standard Deviation	**1549.651**	**1337.761**	**1400.989**

Table 5.14.: T-test for the length of the scan path on the overall drawing area

	T-test		
	CL-CSG	**CL-FCL**	**CSG-FCL**
Z	-1.250b	-0.966b	-0.738c
p	0.211	0.334	0.460

Table 5.15.: Scan path duration in seconds on the overall drawing area

Scan Path Duration			
ID	**CL**	**CSG**	**FCL**
0584134975	9.430	8.961	11.764
0761924407	6.855	6.387	8.898
0935711842	2.550	1.910	1.940
1228890554	5.191	4.497	6.091
1325908148	12.287	7.753	6.301
1921419896	6.193	4.939	5.564
2415213624	9.921	8.874	10.147
318108245	4.789	12.164	4.830
3723364039	4.441	3.308	4.611
4430417015	4.447	6.946	4.628
6477520522	5.144	4.251	5.148
8593499818	5.765	3.729	9.120
9626467901	4.273	6.416	8.826
9754525723	7.035	10.243	8.402
9849337422	8.376	12.469	8.255
Average	**6.446**	**6.856**	**6.968**
Standard Deviation	**2.588**	**3.204**	**2.619**

Table 5.16.: T-test for the duration of the scan path on the overall drawing area

	T-test		
	CL-CSG	**CL-FCL**	**CSG-FCL**
Z	-0.057b	-1.533b	-0.511b
p	0.955	0.125	0.609

Scan path duration

As illustrated in Table 5.15, the *CL* obtained the best performance in terms of the scan path duration. It had an average time of 6.446 seconds, followed by the *CSG* with 6.856 and the *FCL* with 6.968. This result indicates that the drawing is processed faster using the *CL*. Nevertheless, the difference was found not to be statistically significant as depicted in Table 5.16.

Spatial density

As shown in Table 5.17, the *CSG* obtained the best performance in terms of the spatial density. It had an average of 0.355, followed by the *FCL* with 0.383 and the *CL* with 0.437. This result suggests that the *CSG* offers the most direct visual search. Furthermore, the difference was found to be statistically significant as shown in Table 5.18. The *CSG* offered a more direct search in comparison to the *CL* ($[Z = -3.413b, p = 0.001]$) and the *FCL* ($[Z = 2.212c, p = 0.027]$). Similarly, the *FCL* offered a more direct search in comparison to the *CL* ($[Z = -3.071b, p = 0.002]$).

Table 5.17.: Spatial density on the overall drawing area

Spatial Density			
ID	**CL**	**CSG**	**FCL**
0584134975	0.484	0.366	0.394
0761924407	0.426	0.310	0.402
0935711842	0.432	0.376	0.374
1228890554	0.426	0.278	0.374
1325908148	0.500	0.412	0.428
1921419896	0.456	0.352	0.354
2415213624	0.444	0.366	0.484
318108245	0.410	0.374	0.352
3723364039	0.366	0.348	0.382
4430417015	0.448	0.410	0.430
6477520522	0.464	0.414	0.352
8593499818	0.402	0.242	0.268
9626467901	0.402	0.324	0.346
9754525723	0.474	0.382	0.438
9849337422	0.416	0.372	0.366
Average	**0.437**	**0.355**	**0.383**
Standard Deviation	**0.036**	**0.049**	**0.051**

Table 5.18.: T-test for the spatial density on the overall drawing area

	T-test		
	CL-CSG	**CL-FCL**	**CSG-FCL**
Z	-3.413b	-3.071b	2.212c
p	0.001	0.002	0.027

Table 5.19.: Fixation/saccade ratio on the overall drawing area

Fixation/Saccade Ratio			
ID	**CL**	**CSG**	**FCL**
0584134975	13.730	13.931	15.587
0761924407	10.163	10.545	10.559
0935711842	1.936	1.770	3.956
1228890554	6.873	6.057	11.742
1325908148	8.902	15.608	9.731
1921419896	10.966	10.835	12.987
2415213624	8.307	11.342	6.533
318108245	11.603	16.821	14.303
3723364039	3.986	5.211	3.703
4430417015	7.907	7.193	7.450
6477520522	5.157	5.748	6.197
8593499818	9.133	3.522	13.154
9626467901	9.016	11.329	14.747
9754525723	6.127	9.941	4.531
9849337422	12.497	11.807	16.710
Average	**8.420**	**9.444**	**10.126**
Standard Deviation	**3.228**	**4.398**	**4.459**

Table 5.20.: T-test for the fixation/saccade ratio on the overall drawing area

	T-test		
	CL-CSG	**CL-FCL**	**CSG-FCL**
Z	-1.250b	-2.443b	-0.682b
p	0.211	0.015	0.496

Fixation/saccade ratio

As displayed in Table 5.19, the *FCL* obtained the highest value in terms of the fixation/saccade ratio. It had an average of 10.126, followed by the *CSG* with 9.444 and the *CL* with 8.420. This result suggests that the students required more time to process the drawing using the *FCL* as it is confirmed by the duration of the scan paths. However, the difference was found not to be statistically significant as shown in Table 5.20.

Eye-tracking metrics on the areas of interest

The AOIs were placed according to the positioning rules of the dynamic graph drawings. For example, the *CL* always changed the position of the actors during the exploration of the dynamic network. Therefore, it was necessary to adjust the AOIs for each snapshot in the sequence as displayed in Figures 5.17, 5.18, 5.19, 5.20 and 5.21. The *CSG* and the *FCL* always maintained the actors in the same position during the temporal navigation. As depicted in Figures 5.22, 5.23, 5.24, 5.25, 5.26, 5.27, 5.28, 5.29, 5.30 and 5.31, it was not necessary to adjust the AOIs.

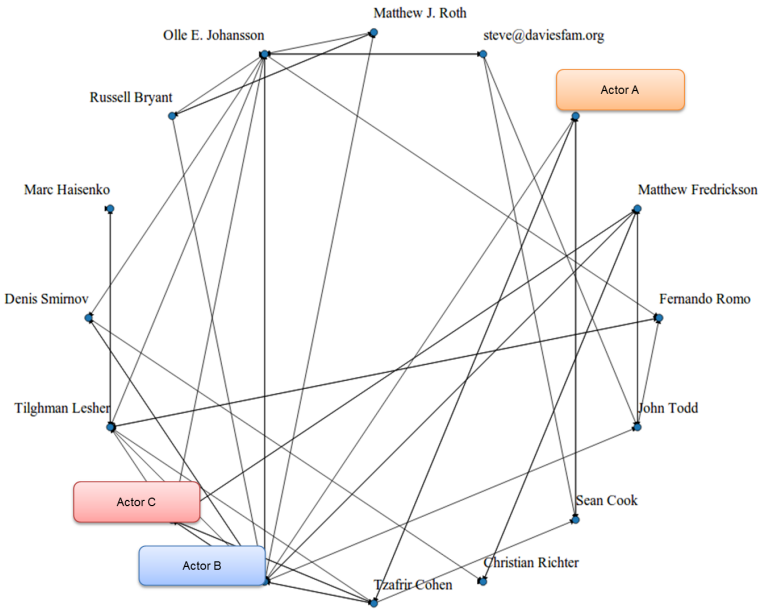

Figure 5.17.: Areas of interest in the circular layout for snapshot 1

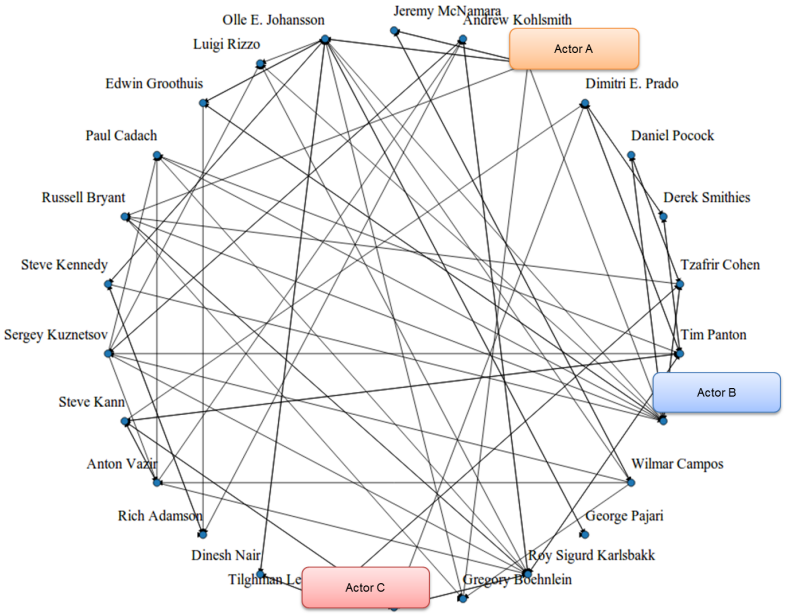

Figure 5.18.: Areas of interest in the circular layout for snapshot 2

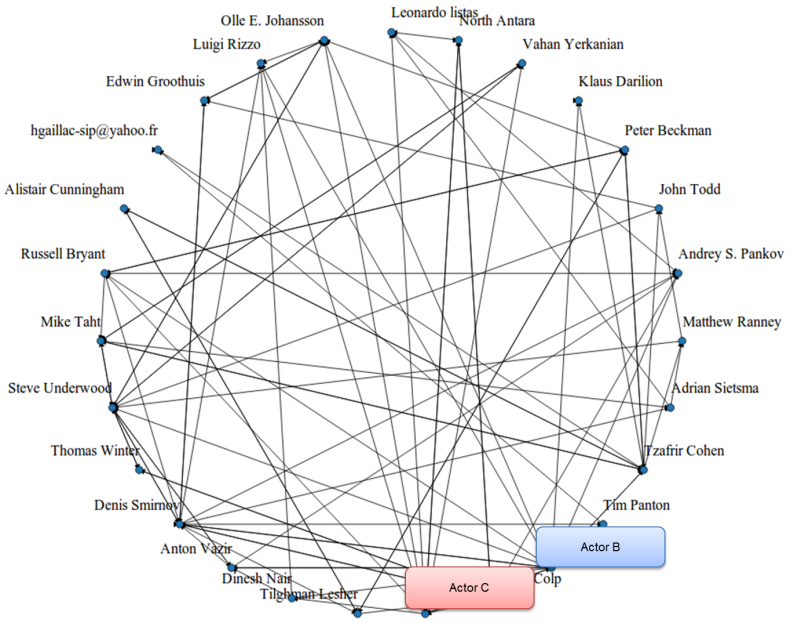

Figure 5.19.: Areas of interest in the circular layout for snapshot 3

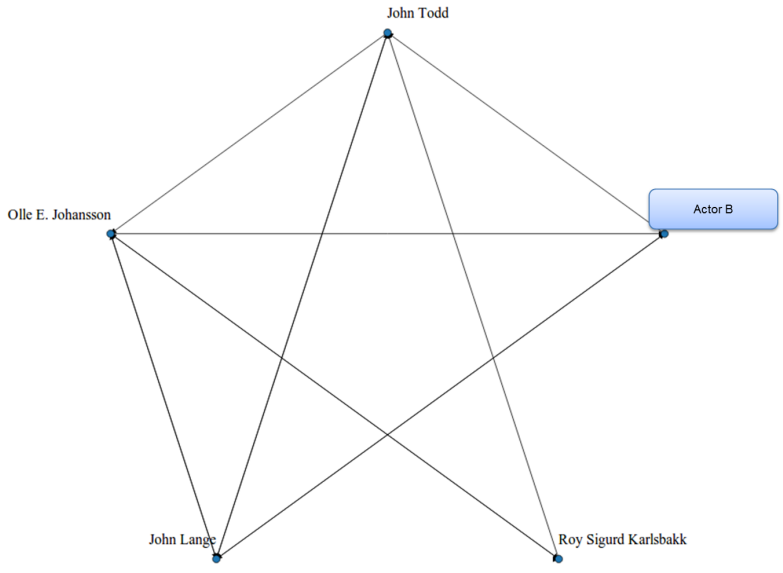

Figure 5.20.: Areas of interest in the circular layout for snapshot 4

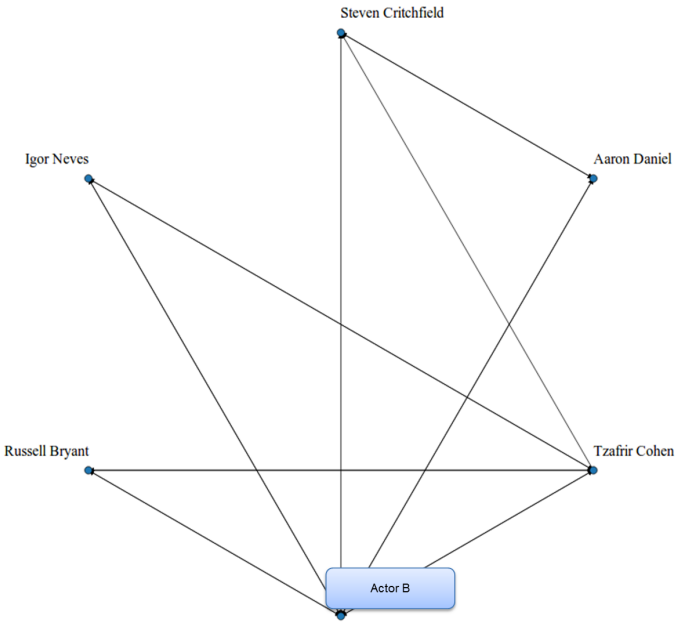

Figure 5.21.: Areas of interest in the circular layout for snapshot 5

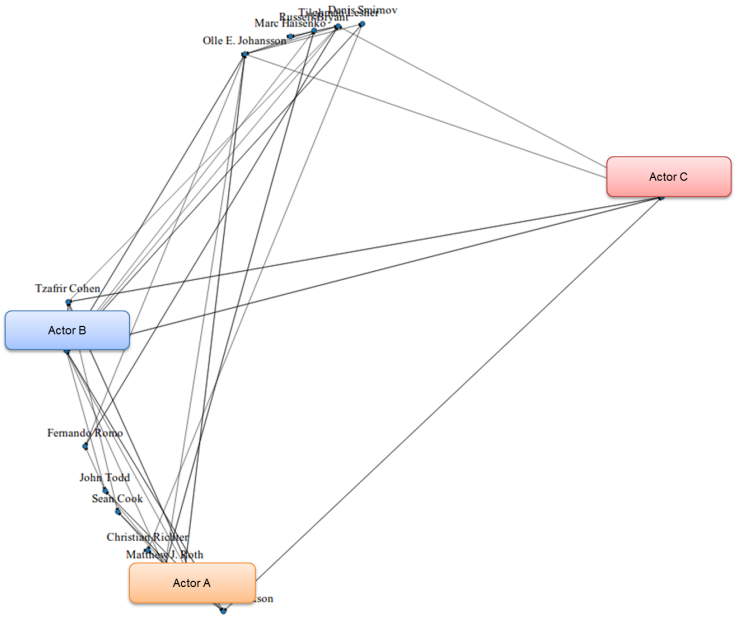

Figure 5.22.: Areas of interest in the circular super graph for snapshot 1

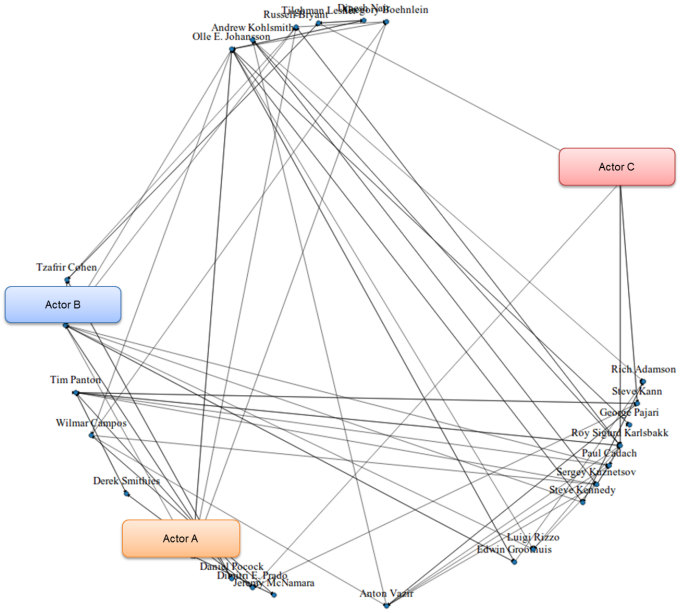

Figure 5.23.: Areas of interest in the circular super graph for snapshot 2

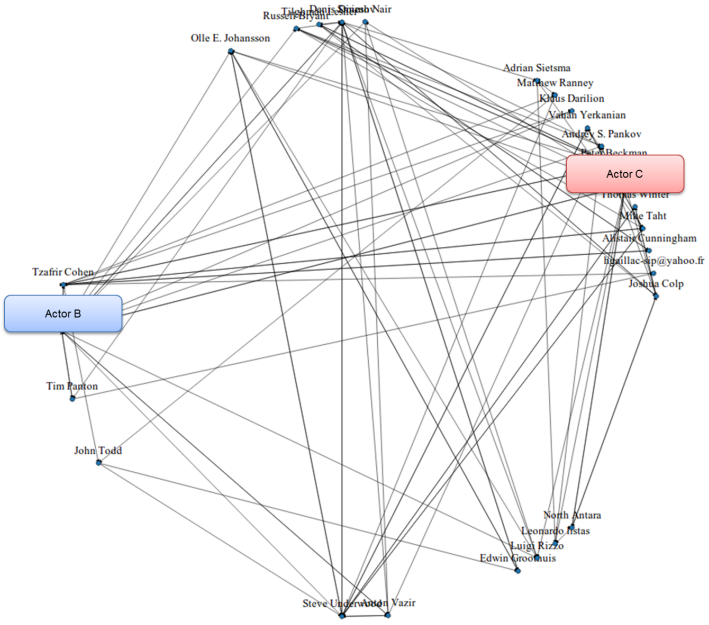

Figure 5.24.: Areas of interest in the circular super graph for snapshot 3

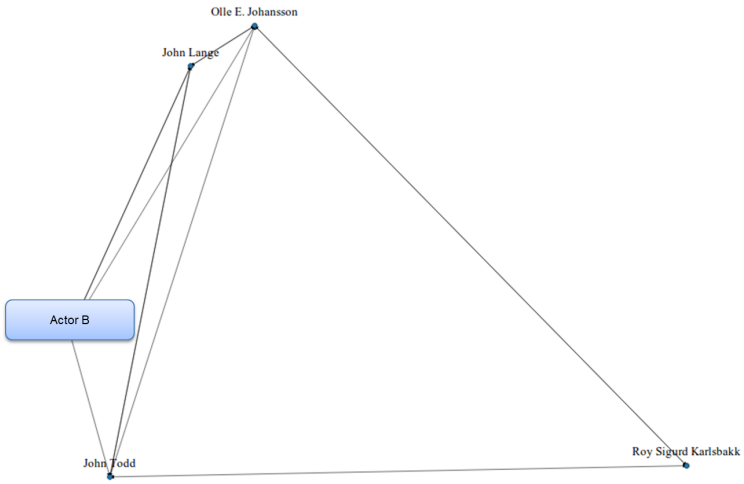

Figure 5.25.: Areas of interest in the circular super graph for snapshot 4

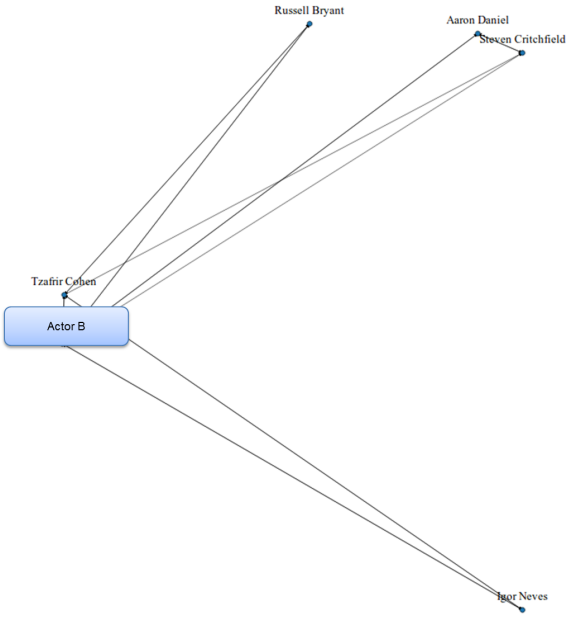

Figure 5.26.: Areas of interest in the circular super graph for snapshot 5

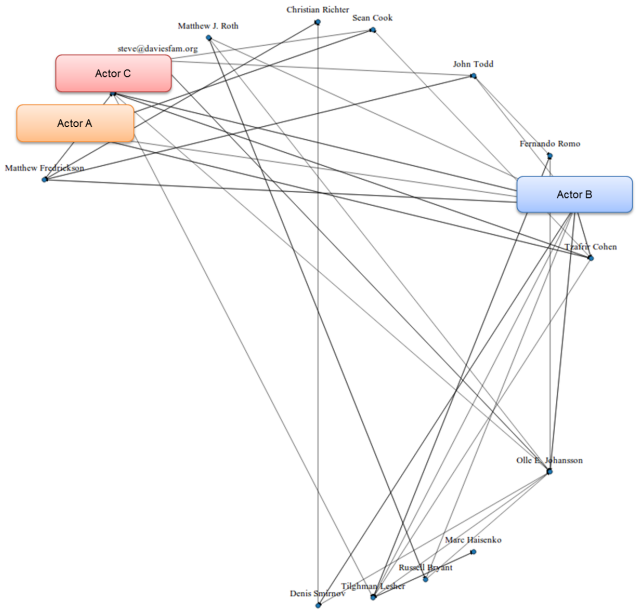

Figure 5.27.: Areas of interest in the foresighted circular layout for snapshot 1

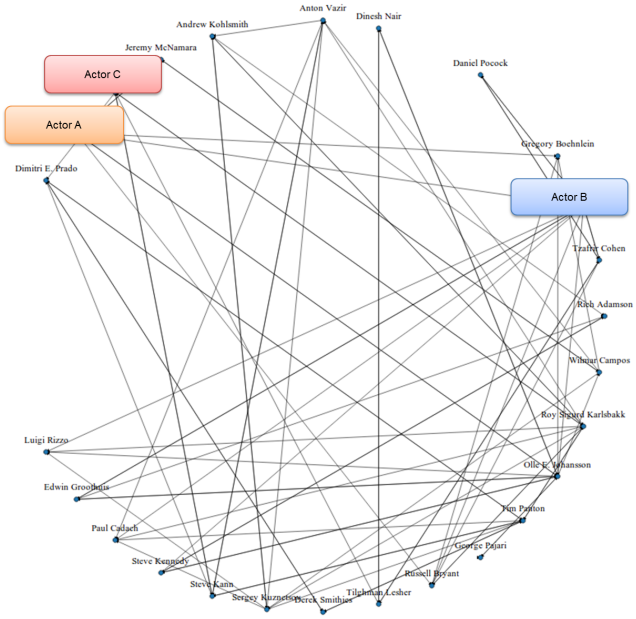

Figure 5.28.: Areas of interest in the foresighted circular layout for snapshot 2

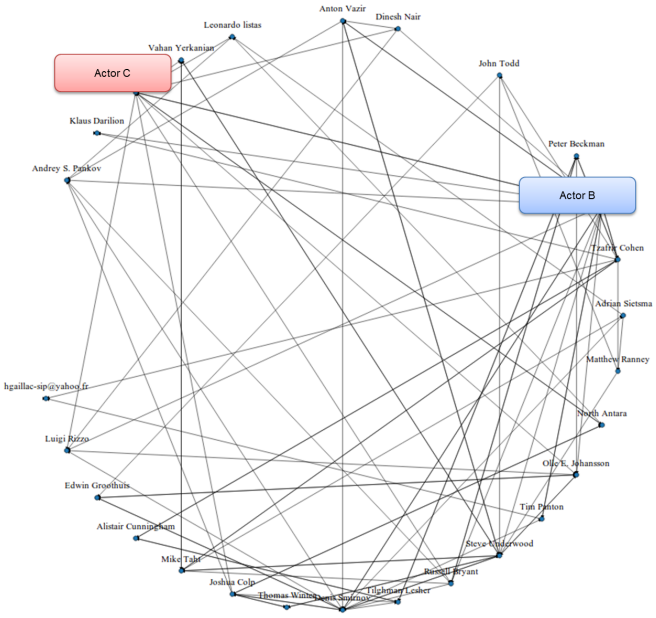

Figure 5.29.: Areas of interest in the foresighted circular layout for snapshot 3

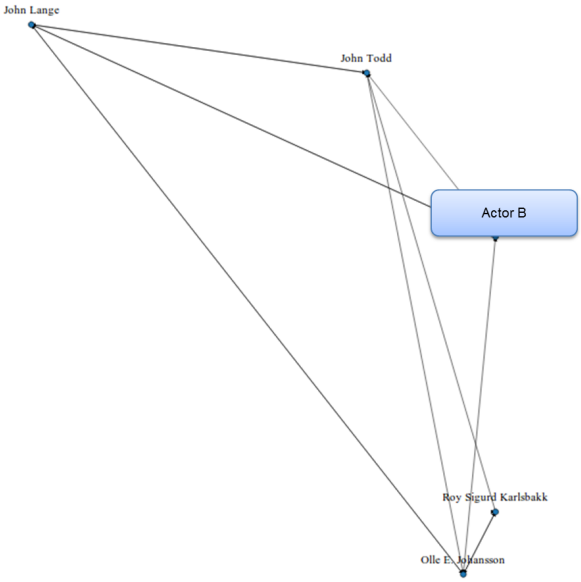

Figure 5.30.: Areas of interest in the foresighted circular layout for snapshot 4

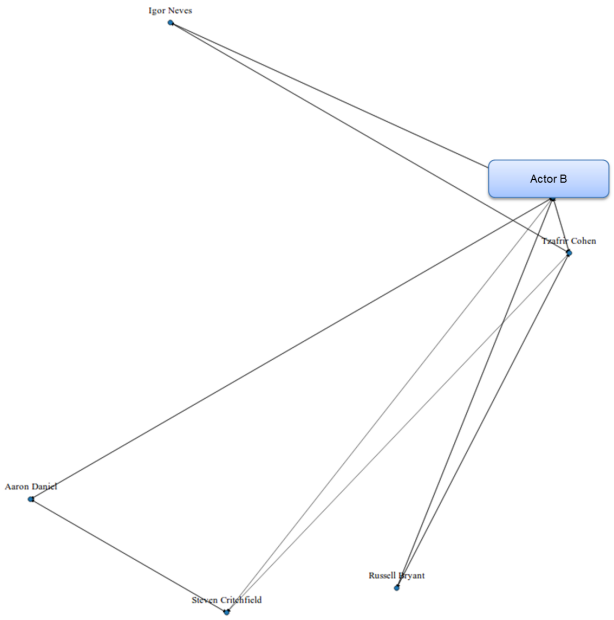

Figure 5.31.: Areas of interest in the foresighted circular layout for snapshot 5

Fixations on target

As shown in Table 5.21, the *CSG* obtained the best results in terms of the fixations on target for *Actor A*. It had an average of 4.4% (0.044) of the fixations on target, followed by the *FCL* with 4.0% (0.040) and the *CL* with 2.6% (0.026). This indicates that the *CSG* offered the most accurate visual search to locate *Actor A*. However, the difference was found not to be statistically significant as displayed in Table 5.22.

Table 5.23 illustrates that the *CSG* obtained the best results in terms of the fixations on target for *Actor B*. It had an average of 9.7% (0.097) of the fixations on target, followed by the *CL* with 5.9% (0.059) and the *FCL* with 4.9% (0.049). This suggests that the *CSG* offered the most accurate visual search to locate *Actor B*. Nonetheless, the difference was found not to be statistically significant as depicted in Table 5.24.

As shown in Table 5.25, the *FCL* obtained the best results in terms of the fixations on target for *Actor C*. It had an average of 3.5% (0.035) of the fixations on target, followed by the *CSG* with 3.1% (0.031) and the *CL* with 1.1% (0.011). This indicates that the *FCL* offered the most accurate visual search to locate *Actor C*. Nevertheless, the difference was found not to be statistically significant as emphasized in Table 5.26.

Time to first fixation

Considering Table 5.27, the *CL* obtained the best results in terms of the time to first fixation for *Actor A*. It had an average time of 1.095 seconds, followed by the *FCL* with 1.365 seconds and the *CSG* with 1.594 seconds. This suggests that *Actor A* was located faster using the *CL*. However, the difference was found not to be statistically significant as shown in Table 5.28.

Table 5.21.: Fixations on target for actor A

Fixations on Target - Actor A			
ID	**CL**	**CSG**	**FCL**
0584134975	0.034	0.052	0.078
0761924407	0.037	0.000	0.027
0935711842	0.026	0.080	0.035
1228890554	0.025	0.000	0.033
1325908148	0.030	0.028	0.060
1921419896	0.032	0.042	0.048
2415213624	0.042	0.077	0.018
318108245	0.052	0.040	0.046
3723364039	0.008	0.060	0.031
4430417015	0.024	0.045	0.040
6477520522	0.006	0.050	0.054
8593499818	0.000	0.100	0.039
9626467901	0.021	0.006	0.053
9754525723	0.014	0.047	0.000
9849337422	0.034	0.035	0.037
Average	**0.026**	**0.044**	**0.040**
Standard Deviation	**0.014**	**0.029**	**0.018**

Table 5.22.: T-test for the fixations on target appearing over actor A

	T-test		
	CL-CSG	**CL-FCL**	**CSG-FCL**
Z	-1.931b	-2.329b	-0.443c
p	0.053	0.020	0.658

Table 5.23.: Fixations on target for actor B

Fixations on Target - Actor B			
ID	**CL**	**CSG**	**FCL**
0584134975	0.069	0.089	0.102
0761924407	0.071	0.049	0.000
0935711842	0.051	0.103	0.035
1228890554	0.054	0.000	0.057
1325908148	0.067	0.115	0.043
1921419896	0.097	0.102	0.096
2415213624	0.056	0.157	0.004
318108245	0.078	0.100	0.109
3723364039	0.000	0.145	0.038
4430417015	0.097	0.161	0.129
6477520522	0.059	0.200	0.033
8593499818	0.063	0.077	0.007
9626467901	0.029	0.021	0.016
9754525723	0.025	0.023	0.000
9849337422	0.064	0.115	0.066
Average	**0.059**	**0.097**	**0.049**
Standard Deviation	**0.026**	**0.057**	**0.043**

Table 5.24.: T-test for the fixations on target appearing over actor B

	T-test		
	CL-CSG	**CL-FCL**	**CSG-FCL**
Z	-2.387b	-0.853c	-2.483c
p	0.017	0.394	0.013

Table 5.25.: Fixations on target for actor C

Fixations on Target - Actor C			
ID	**CL**	**CSG**	**FCL**
0584134975	0.004	0.032	0.019
0761924407	0.016	0.005	0.025
0935711842	0.006	0.060	0.013
1228890554	0.012	0.000	0.046
1325908148	0.012	0.033	0.050
1921419896	0.003	0.026	0.055
2415213624	0.005	0.068	0.003
318108245	0.034	0.057	0.064
3723364039	0.000	0.071	0.024
4430417015	0.024	0.057	0.021
6477520522	0.019	0.054	0.022
8593499818	0.007	0.000	0.093
9626467901	0.000	0.000	0.046
9754525723	0.010	0.002	0.012
9849337422	0.018	0.008	0.033
Average	**0.011**	**0.031**	**0.035**
Standard Deviation	**0.010**	**0.028**	**0.024**

Table 5.26.: T-test for the fixations on target appearing over actor C

	T-test		
	CL-CSG	**CL-FCL**	**CSG-FCL**
Z	-2.372b	-2.655b	-0.126b
p	0.018	0.008	0.900

Table 5.27.: Time to first fixation for actor A

Time to First Fixation - Actor A			
ID	**CL**	**CSG**	**FCL**
0584134975	1.950	1.411	0.261
0761924407	1.330	0.000	1.051
0935711842	0.644	2.375	1.231
1228890554	0.794	0.000	1.090
1325908148	2.942	3.541	0.772
1921419896	0.509	1.690	0.301
2415213624	0.996	1.642	0.146
318108245	0.992	3.228	0.670
3723364039	0.196	0.478	4.251
4430417015	1.641	2.221	4.411
6477520522	2.202	0.882	0.411
8593499818	0.000	1.089	3.223
9626467901	0.043	1.578	1.959
9754525723	0.502	2.310	0.000
9849337422	1.692	1.461	0.693
Average	**1.095**	**1.594**	**1.365**
Standard Deviation	**0.854**	**1.039**	**1.451**

Table 5.28.: T-test for the time to first fixation over actor A

	T-test		
	CL-CSG	**CL-FCL**	**CSG-FCL**
Z	-1.590b	-0.114b	-0.795c
p	0.112	0.910	0.427

As shown in Table 5.29, the *CSG* obtained the best results in terms of the time to first fixation for *Actor B*. It had an average time of 2.347 seconds, followed by the *FCL* with 3.078 seconds and the *CL* with 3.175 seconds. This emphasizes that *Actor B* was located faster using the *CSG*. Nonetheless, the difference was found not to be statistically significant as depicted in Table 5.30.

Table 5.31 clarifies that the *CL* obtained the best results in terms of the time to firs fixation for *Actor C*. It had an average time of 1.688 seconds, followed by the *FCL* with 2.019 seconds and the *CSG* with 3.283 seconds. This suggests that *Actor C* was located faster using the *CL*. Nevertheless, the difference was found not to be statistically significant as shown in Table 5.32.

Discussion

In order to get a further insight on how the visual stability of a dynamic graph drawing affects the search efficiency and user experience, Spearman's rank correlation [110]

Table 5.29.: Time to first fixation for actor B

Time to First Fixation - Actor B			
ID	**CL**	**CSG**	**FCL**
0584134975	6.191	1.916	4.787
0761924407	3.085	2.256	0.000
0935711842	3.237	2.015	4.050
1228890554	3.846	0.000	0.282
1325908148	4.667	6.394	2.527
1921419896	1.983	2.299	4.428
2415213624	2.869	1.515	2.550
318108245	4.129	4.148	2.892
3723364039	0.000	1.766	5.958
4430417015	1.868	3.905	11.444
6477520522	4.441	0.809	0.278
8593499818	2.119	2.332	2.425
9626467901	3.004	2.446	1.227
9754525723	2.075	1.770	0.000
9849337422	4.110	1.637	3.322
Average	**3.175**	**2.347**	**3.078**
Standard Deviation	**1.484**	**1.514**	**2.965**

Table 5.30.: T-test for the time to first fixation over actor B

	T-test		
	CL-CSG	**CL-FCL**	**CSG-FCL**
Z	-1.306b	-0.909b	-0.795c
p	0.191	0.363	0.427

Table 5.31.: Time to first fixation for actor C

Time to First Fixation - Actor C			
ID	**CL**	**CSG**	**FCL**
0584134975	5.951	2.762	0.617
0761924407	2.153	0.046	7.681
0935711842	2.274	6.658	6.598
1228890554	1.528	0.000	1.609
1325908148	1.406	10.844	1.257
1921419896	0.709	4.161	2.847
2415213624	0.543	1.729	0.123
318108245	2.170	3.014	0.715
3723364039	0.000	2.744	2.091
4430417015	1.792	8.065	3.872
6477520522	3.889	4.212	0.286
8593499818	1.328	0.000	0.723
9626467901	0.000	0.000	1.078
9754525723	0.792	4.065	0.086
9849337422	0.789	0.946	0.701
Average	**1.688**	**3.283**	**2.019**
Standard Deviation	**1.550**	**3.218**	**2.333**

Table 5.32.: T-test for the time to first fixation over actor C

	T-test		
	CL-CSG	**CL-FCL**	**CSG-FCL**
Z	-1.601b	-0.341b	-1.647c
p	0.109	0.733	0.100

Table 5.33.: Results of the correlation analysis for the CL

		Correlation Analysis - CL				
		C1	**C2**	**C3**	**C4**	**C5**
Number of fixations	ρ	0.130	0.062	-0.022	0.068	0.170
	p	0.643	0.825	0.937	0.810	0.545
Duration of fixations	ρ	0.084	0.238	-0.250	-0.019	-0.354
	p	0.767	0.393	0.369	0.947	0.195
Number of saccades	ρ	0.158	-0.111	0.109	0.122	0.371
	p	0.573	0.694	0.698	0.666	0.173
Duration of saccades	ρ	-0.089	0.035	0.472	-0.230	**0.539**
	p	0.751	0.902	0.076	0.410	**0.038**
Saccadic amplitude	ρ	-0.186	0.027	-0.148	-0.115	-0.149
	p	0.506	0.923	0.599	0.684	0.595
Scan path length	ρ	0.121	-0.106	0.096	**0.516**	-0.118
	p	0.667	0.706	0.733	**0.049**	0.675
Scan path duration	ρ	0.196	0.026	-0.002	0.401	-0.249
	p	0.485	0.928	0.995	0.139	0.371
Spatial density	ρ	-0.016	-0.052	0.111	0.189	0.170
	p	0.955	0.853	0.693	0.501	0.545
Fixations/Saccade ratio	ρ	0.076	0.071	-0.400	0.041	**-0.522**
	p	0.787	0.800	0.140	0.884	**0.046**

(Spearman's ρ) was utilized as a measure of statistical dependence between the eye-tracking metrics and the questionnaires.

According to the model of visual stability, the *Circular Layout* (CL) presented the highest values of graph drawing offset (GDO) because the actors were constantly changing their position in the Euclidean Space. Such a characteristic had a negative impact on the users, who reported that it was difficult to track the requested actors in the dynamic network. In addition, the most extensive search patterns were found with the *CL*. The length of the scan path was found to be positively correlated with the criterion C4 ($\rho = 0.516, p < 0.05$), suggesting that longer scan paths influence the difficulty to locate an actor. The duration of the saccades was also found to be positively correlated with the criterion C5 ($\rho = 0.539, p < 0.05$), indicating that the distraction produced by the addition and removal of elements from the dynamic graph drawing increases the duration of the saccades. Furthermore, the fixation/saccade ratio was negatively correlated to the criterion C5 ($\rho = -0.522, p < 0.05$). In this case, the distraction produced by the addition and removal of elements from the canvas negatively affected the efficiency of the visual search. The results of the correlation analysis for the *CL* are presented in Table 5.33.

The *Circular Super Graph* (CSG) assigned to every actor in the dynamic network a fixed position in the Euclidean Space. As a result, the graph drawing offset was reduced to zero (GDO = 0). Such a characteristic had a positive impact on the users, who reported it was easier to locate the requested actors in the dynamic network. This was noticed in the eye-tracking metrics. The number of fixations on target presented the highest values among the three dynamic graph drawings, indicating an accurate visual search [56]. Nonetheless, the duration of the fixations was the longest of all drawings subject

Table 5.34.: Results of the correlation analysis for the CSG

		Correlation Analysis - CSG				
		C1	**C2**	**C3**	**C4**	**C5**
Number of fixations	ρ	0.152	-0.093	-0.378	0.324	-0.163
	p	0.588	0.742	0.165	0.239	0.562
Duration of fixations	ρ	0.029	-0.367	0.277	0.107	0.472
	p	0.917	0.179	0.318	0.704	0.076
Number of saccades	ρ	0.152	-0.080	-0.385	0.340	-0.163
	p	0.588	0.776	0.156	0.215	0.562
Duration of saccades	ρ	-0.134	0.177	-0.190	-0.215	0.152
	p	0.635	0.528	0.498	0.443	0.589
Saccadic amplitude	ρ	0.238	-0.078	0.343	-0.107	0.185
	p	0.393	0.781	0.211	0.704	0.509
Scan path length	ρ	0.319	-0.489	-0.234	0.215	-0.139
	p	0.247	0.065	0.401	0.443	0.622
Scan path duration	ρ	0.211	-0.456	-0.148	0.211	0.039
	p	0.451	0.088	0.600	0.451	0.891
Spatial density	ρ	0.272	0.028	-0.489	0.203	-0.231
	p	0.326	0.920	0.064	0.469	0.407
Fixations/Saccade ratio	ρ	0.075	-0.356	0.166	0.178	0.227
	p	0.790	0.193	0.554	0.527	0.415

to evaluation. This suggests that more time is required to process an element [56]. The results of the correlation analysis for the *CSG* are shown in Table 5.34.

The *Foresighted Circular Layout* (FCL) assigned multiple actors to the same position in the Euclidean Space at different points in time. As a result, the graph drawing offset was also reduced to zero (GDO = 0). Such a characteristic had a positive impact on the users, who reported it was easy to locate the requested actors in the dynamic network. Nevertheless, observing multiple actors in the same position at different points in time was found to be confusing as noticed in the eye-tracking metrics. The duration of the fixations presented a decrease in comparison to the *CSG*, indicating that the elements in the drawing are processed faster. In contrast, the length of the scan paths ($\rho = 0.686, p < 0.05$), the scan path duration ($\rho = 0.592, p < 0.05$) and the saccadic amplitude ($\rho = 0.649, p < 0.05$) were found to be positively correlated with the criterion C1. According to Goldberg [56], longer scan paths are an indicator of a more extensive visual search. This suggests although the elements in the drawing are processed faster, the students performed more searches because several actors occupied the same position at different points in time. Moreover, the fixation/saccade ratio was found to be positively correlated with the criterion C2 ($\rho = 0.598, p < 0.05$), indicating that the users perceived the characteristics of the *FCL* as useful, despite its limitations. This was also confirmed by the negative correlation of the criterion C2 with the duration of the saccades ($\rho = -0.636, p < 0.05$). The results of the correlation analysis for the *FCL* are provided in Table 5.35.

Table 5.35.: Results of the correlation analysis for the FCL

		Correlation Analysis - FCL				
		C1	**C2**	**C3**	**C4**	**C5**
Number of fixations	ρ	0.028	-0.247	-0.269	0.418	-0.070
	p	0.921	0.375	0.332	0.121	0.804
Duration of fixations	ρ	0.056	0.256	-0.251	-0.152	0.201
	p	0.842	0.356	0.367	0.589	0.473
Number of saccades	ρ	0.056	-0.323	-0.151	0.388	-0.040
	p	0.842	0.240	0.590	0.153	0.887
Duration of saccades	ρ	-0.338	**-0.636**	0.312	0.218	0.384
	p	0.217	**0.011**	0.258	0.434	0.158
Saccadic amplitude	ρ	**0.649**	0.085	0.328	0.294	-0.134
	p	**0.009**	0.762	0.232	0.287	0.633
Scan path length	ρ	**0.686**	0.304	-0.183	0.044	-0.318
	p	**0.005**	0.271	0.515	0.877	0.249
Scan path duration	ρ	**0.592**	0.313	-0.100	0.076	-0.070
	p	**0.020**	0.255	0.724	0.788	0.804
Spatial density	ρ	0.245	-0.285	-0.326	0.211	-0.407
	p	0.379	0.302	0.236	0.450	0.132
Fixations/Saccade ratio	ρ	0.075	**0.598**	-0.229	-0.372	-0.043
	p	0.790	**0.018**	0.412	0.172	0.879

5.7. Summary

This chapter presented the case study to evaluate the mathematical model of visual stability described in Chapter 4. For this purpose, a group of 15 students was requested to track three actors in three different scenarios. Each one of them illustrated a dynamic network with a dynamic graph drawing whose characteristics were related to the theories of visual stability presented in Chapter 2. An eye-tracking device was used to record their eye movements; a questionnaire was provided to gather information about their experience and the model-based metrics quantified the visual stability of the dynamic graph drawings.

Scenario 1 used a *Circular Layout* (CL) because it provided a drawing with a constant shape, but changed the position of the actors during exploration of the dynamic network. *Scenario 2* used a *Circular Super Graph* (CSG) because it assigned to every actor in the dynamic network a fixed position in the Euclidean Space. Nonetheless, it did not have a constant shape. Finally, *Scenario 3* used a *Foresighted Circular Layout* (FCL) because it provided a drawing with the minimal number of changes during the exploration of the dynamic network. However, it allowed multiple actors to occupy the same position at different points in time.

The dynamic network used in the case study was extracted from the developer mailing list of the Asterisk framework and had five snapshots. Each one represented a month of communication between the developers [126]. In the drawings, the developers were represented as nodes and an e-mail exchange as an edge. The research hypothesis for

the case study implied that *visually stable drawings improve the efficiency of the visual search and the user experience when tracking actors in a dynamic network.*

The data collected from the questionnaires provided a better insight about the user experience. They evaluated five criteria of the dynamic graph drawings, such as the drawing displayed the specific actors in the same position; the drawing technique was useful to track actors in a dynamic network; the changes in the drawing were not noticeable; it was easy to locate an actor on the drawing and the addition or removal of actors and relations from the drawing was not distracting. This information was further analyzed using a Likert scale [77] and a card sorting technique [3].

In addition, the information recorded by the eye-tracking device was analyzed with metrics that have proven to quantify the efficiency of the visual search [26, 56, 57, 94]. Two settings were proposed for the analysis. The first applied the metrics to the overall drawing area, while the second one where the three actors were located. These were the areas of interest (AOIs).

The results obtained suggest that the visual stability of a dynamic graph drawing affects the efficiency of the visual search and the user experience when tracking actors in a dynamic network. Dynamic graph drawings changing the position of the nodes to achieve a constant shape present an extensive visual search, which difficult the tracking tasks. As an alternative, dynamic graph drawings can assign to every actor in the dynamic network a fixed position in the Euclidean Space. This result on a more accurate visual search. However, the constant addition and removal of elements from the dynamic graph drawing affect negatively the user experience. A solution to this problem is to allow multiple actors to occupy the same position at different points in time. This reduces the constant addition and removal of elements, which improves the user experience at the cost of losing accuracy on the visual search. In the subsequent chapters, the mathematical model of visual stability is used to evaluate new dynamic graph drawing algorithms. These approaches aim to support an efficient visual search along with a satisfying user experience when tracking actors in a dynamic network.

6. Algorithms to support visual stability in dynamic graph drawings

6.1. Introduction

The visual stability of a dynamic graph drawing is an attribute which should be considered when tracking of actors or other network related patterns over time. As it has been discussed in Chapter 5, the drawings that assign the actors a fixed position in the Euclidean Space provides an efficient visual search. However, the user experience is negatively affected due to their visual instability. In contrast, the drawings that assign multiple actors to the same position at different points in time improve the user experience due to their visual stability. Nonetheless, this comes at the cost of losing efficiency in the visual search.

This chapter presents two algorithms to create dynamic graph drawings, aiming to support the user experience and an efficient visual search when tracking actors or patterns in a dynamic network. Both approaches are inspired by the foresighted graph layout [31]. On the one hand, the flickering reduction algorithm is designed to minimize the addition and removal of elements during the temporal navigation. On the other hand, the degree stabilization algorithm is designed to minimize the changes on the number of connections of the nodes appearing in the dynamic graph drawing.

The content of this chapter is based on the publication *"Using Visual Stability to Support Search Efficiency and User Experience in Dynamic Graph Drawings"* [101].

6.2. Basis of the algorithms

The developed algorithms rely on a structure called *lifetime matrix* or *LTM*. It contains the information about the appearance of the actors in a dynamic social network. The *LTM* is defined as follows:

- Let a dynamic graph/network G be a sequence $G = [g^1, g^2, g^3, ..., g^n]$ of graphs with $g^j = (V^j, E^j)$.
- Let all the actors in the dynamic graph/social network be defined by the set $A = \{\cup_{j=1}^{n} V(g^j)\}$

Therefore, the LTM is a matrix in the form:

$$\texttt{LTM} = \begin{bmatrix} a^{11} & a^{12} & a^{13} & a^{14} & ... & a^{1n} \\ a^{21} & a^{22} & a^{23} & a^{24} & ... & a^{2n} \\ a^{31} & a^{32} & a^{33} & a^{34} & ... & a^{3n} \\ a^{41} & a^{42} & a^{43} & a^{44} & ... & a^{4n} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ a^{m1} & a^{m2} & a^{m3} & a^{m4} & ... & a^{mn} \end{bmatrix}$$

where $m = |A|$, $n = |G|$. An element a^{ij} of the LTM represents a property of an actor i in the time j and it is assigned by the developed algorithms. In addition, a single row of the LTM is referred as *lifetime vector* or *LTV*.

6.3. Flickering reduction algorithm

The nodes of a dynamic graph drawing can be considered as "bulbs", which alternate between the presence and absence of light. The change in their luminosity depends on the lifetime of the actors in the dynamic network. Actors with a longer lifetime illuminate the bulbs for longer periods of time, while actors with a shorter lifetime do the opposite. Unfortunately, the lifetime of the actors tends to be discontinuous, making it impossible to maintain the bulbs constantly illuminated. As a result, the visual stability of the dynamic graph drawing is affected by a "flickering".

The *flickering reduction algorithm* was developed to minimize such an anomaly from the dynamic graph drawing. It assumes that the "flickering" is produced by those actors with discontinuous lifetimes and uses them as the starting point to achieve visual stability. In a first instance, the algorithm initializes all the positions a^{ij} of the LTM by placing 1 to indicate the presence of the actor i in time j or 0 to indicate his absence. Then, it detects the actors with discontinuous lifetimes employing one of its two indicators. On the one hand, the *flickering index* refers to the number of times an actor changes from an appearance to a disappearance and vice versa over all the periods of time in the dynamic network. Figure 6.1 illustrates the notion of the *flickering index*, while Figure 6.2 illustrates how to calculate it using the LTV of an actor. On the other hand, the *gap index* refers to the number of inactive time periods an actor presents in the dynamic network. Figure 6.3 presents the notion behind the *gap index* and Figure 6.4 illustrates how to calculate it using the LTV of an actor. The selected index is used to build a *priority queue*, which contains the "flickering" produced by each actor in the dynamic network ordered from the highest value to the lowest. Figure 6.5, outlines the function to build the priority queue.

Once the priority queue has been built, the algorithm proceeds to compute a node partition as in the foresighted graph layout [31]. It selects the actor with the highest index from the priority queue. Then, a new component is instantiated and the selected actor is placed in this location. Consequently, the new component is affected by the "flickering". Therefore, the algorithm needs to find other actors that can be placed in the same position to reduce the anomaly. The main condition is that an actor must have

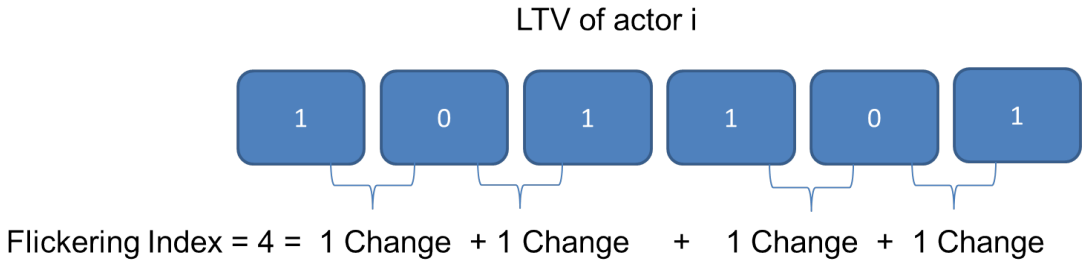

Figure 6.1.: Notion of the flickering index

```
1: function CALCULATEFLICKERINGINDEX(LTV)
2:     int flickeringIndex ← 0;
3:     for i ← 0, LTV.length do
4:         if i + 1 < LTV.length and LTV[i]! = LTV[i + 1] then
5:             flickeringIndex ← flickeringIndex + 1;
6:         end if
7:     end for
8:     return flickeringIndex;
9: end function
```

Figure 6.2.: Pseudo-code to calculate the flickering index of an actor using the LTV

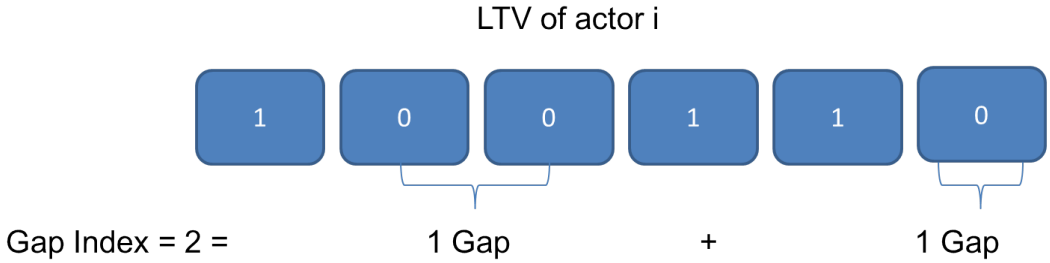

Figure 6.3.: Notion of the gap index

```
1: function CALCULATEGAPINDEX(LTV)
2:     int gapIndex ← 0;
3:     for i ← 0, LTV.length do
4:         if LTV[i]! = 1 then
5:             for j ← i, LTV.length do
6:                 if LTV[j]! = 0 then
7:                     gapIndex ← gapIndex + 1;
8:                     i ← j;
9:                     break;
10:                end if
11:            end for
12:        end if
13:    end for
14:    return gapIndex;
15: end function
```

Figure 6.4.: Pseudo-code to calculate the gap index of an actor using the LTV

```
1: function buildPriorityQueue(LTM)
2:     int[] prioritQueue ← new int[LTM.rows];
3:     for i ← 0, LTM.rows do
4:         int[] ltv ← LTM[i];
5:         int index ← calculateSelectedIndex(ltv);
6:         priorityQueue[i] ← index;
7:     end for
8:     sort(priorityQueue);
9:     return priorityQueue;
10: end function
```

Figure 6.5.: Pseudo-code used to build the priority queue

a disjoint lifetime with all the actors inserted into the component. If this condition is satisfied, the actor is placed in the component. Otherwise, the search process continues. In case it is not possible to find useful candidates, the actors in the component are marked as "processed". These actors are not allowed to be selected again as candidates since they have already been used to reduce the "flickering". The aforementioned steps are repeated until all the elements in the propriety queue have been processed. The result of the algorithm is a node partition similar to the one produced by the foresighted graph layout [31], while the edge partition is computed as it is mentioned in the original approach.

6.4. Degree stabilization algorithm

The *degree stabilization algorithm* was developed following a social network analysis perspective. As illustrated in Figure 6.6, it is designed to minimize the changes on the connections of the nodes that appear in the dynamic graph drawing despite their appearance or disappearance at different points in time. In a first instance, the algorithm initializes all the positions a^{ij} of the LTM with the *degree centrality* [117] of the actor i in time j to denote his presence or with -1 to denote his absence.

Once the LTM has been initialized, the algorithm proceeds to compute a node partition as in the foresighted graph layout [31]. An actor is arbitrarily selected and placed into a new component. The selected actor is used to attach two metrics to the new component. On the one hand, the *average degree centrality* stands for the average number of connections a node maintains over time. As shown in Figure 6.7, this metric can be calculated using the LTV of the actor in the component, however considering the average value of those positions in which the degree centrality is different from -1. On the other hand, the *average degree gradient* is the second metric attached to the component. It describes the average number of connections a node loses over time. Likewise, this metric can be calculated using the LTV of the actor in the component, but only considering the average of the differences in the degree centrality for those positions different from -1 as depicted in Figure 6.8.

Subsequently, the algorithm needs to find other actors that can share a position in the same component. These actors must have a disjoint lifetime and must contribute to stabilize the connections of the nodes over time. An actor is considered to contribute if

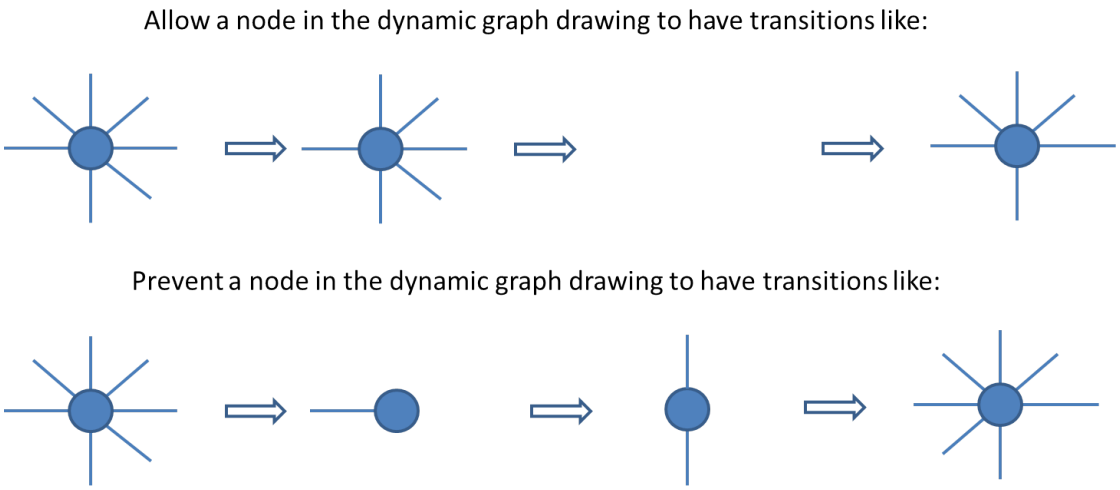

Figure 6.6.: Notion of the degree stabilization algorithm

```
1:  function averageDegree(LTV)
2:      double degree ← 0;
3:      double appearance ← 0;
4:      for i ← 0, LTV.length do
5:          if LTV[i]! = −1 then
6:              degree ← degree + ltv[i];
7:              appearance ← appearance + 1;
8:          end if
9:      end for
10:     return degree/appearance;
11: end function
```

Figure 6.7.: Pseudo-code to calculate the average degree centrality in the degree stabilization algorithm

```
1: function averageGradient(LTV)
2:     int gradient ← 0;
3:     int appearance ← 0;
4:     for i ← 0, LTV.length do
5:         if LTV[i]! = −1 then
6:             appearance ← appearance + 1;
7:             for j ← i, LTV.length do
8:                 if LTV[j]! = −1 then
9:                     appearance ← appearance + 1;
10:                    gradient ← gradient + abs(LTV[i] − LTV[j]);
11:                    i ← j;
12:                    break;
13:                end if
14:            end for
15:        end if
16:    end for
17:    return gradient/appearance;
18: end function
```

Figure 6.8.: Pseudo-code to calculate the average degree gradient in the degree stabilization algorithm

```
1: function contribute(candidateAD, componentAD, componentAG)
2:     if abs(candidateAD − componentAD) <= componentAG then
3:         return true;
4:     end if
5:     return false;
6: end function
```

Figure 6.9.: Pseudo-code to validate the contribution of an actor in the degree stabilization algorithm

the absolute value of the difference between his average degree centrality and the average degree centrality attached to the component is lower or equal than the average gradient attached to the component. Such a validation is illustrated in Figure 6.9. In case the restrictions are not satisfied, the search process continues. Otherwise, the actor is placed in the component. In addition, the algorithm updates the average degree centrality as well as the average degree gradient attached to the component, using the LTVs of all actors stored in this location.

The update procedure makes the average degree gradient to decrease each time a new actor is inserted into the component, while the average degree centrality becomes more stable. As a result, the connections between the nodes in the dynamic graph drawing present fewer changes. Moreover, the component becomes more restrictive and prevents too many actors to occupy the same position at different points in time. In case the algorithm is not able to find viable candidates, the actors in the component are marked as "processed" and are not allowed to be selected again as candidates. These steps are repeated until all actors in the dynamic network have been assigned to a component. The result of the algorithm is a node partition similar to the one produced by the foresighted

graph layout [31]. However, its components only store a few nodes. Regarding the edge partition, it is computed as described in the original foresighted graph layout [31].

6.5. Summary

This chapter presented two different algorithms to create dynamic graph drawings. They aim to support a satisfying user experience, along with an efficient visual search, when tracking actors or other network related patterns over time. The algorithms are inspired by the foresighted graph layout [31], which allows multiple actors to occupy the same node position at different points in time and where the relations between the actors are assigned to the same edge position following the same rule.

On the one hand, the flickering reduction algorithm is designed to minimize the constant addition and removal of elements from a dynamic graph drawing during the temporal navigation. It assumes that such an anomaly is produced by those actors with discontinuous lifetimes and introduces two indicators to detect them. The flickering index stands for the number of times an actor enters and exits the dynamic network, while the gap index stands for the number of inactive time periods an actor presents in the dynamic network. The detected actors are used as the basis to compute a node partition like in the foresighted graph layout [31], whereas the edge partition is computed as it is stated in the original algorithm.

On the other hand, the degree stabilization algorithm is designed to minimize the changes in the connections between the nodes in the dynamic graph drawing. It assumes that such changes are produced by those actors with a high degree gradient. Therefore, the algorithm computes the average degree gradient of all actors in the dynamic network and uses such a metric as the basis to compute a node partition like in the foresighted graph layout [31]. However, there are two conditions that must be fulfilled in order to generate appropriately a node component. First, all the actors inserted into the same node component must have a disjoint lifetime. Second, the connections of these same actors must reduce the average degree gradient. This last restriction prevents a node component to be occupied by a considerable number of actors. In addition, each time an actor is inserted into a node component, the connections of the nodes in the dynamic graph drawing become more stable. The edge partition is computed as it is stated in the foresighted graph layout [31]. In the subsequent chapter of this thesis, the flickering reduction algorithm and the degree stabilization algorithm are evaluated through a case study. The objective is to determine which one provides the best user experience along with the most efficient visual search when tracking actors or other patterns in a dynamic network.

7. Evaluation of the algorithms to support visual stability in dynamic graph drawings

7.1. Introduction

This chapter presents the case study to evaluate the algorithms described in Chapter 6. A group of 20 students participated in the study. They were requested to track three actors in three different dynamic graph drawings. The first one of them was created with the flickering reduction algorithm and the flickering index. The second one with the flickering reduction algorithm and the gap index. The last one of them was generated employing the degree stabilization algorithm. An eye-tracking device was utilized as part of the study. It recorded the eye movements performed by the participants. In addition, a questionnaire was provided to gather information about their experience, while the model-based metrics quantified the visual stability of the dynamic graph drawings.

The results obtained from the study suggest that visually stable dynamic graph drawings provide a satisfying user experience when tracking actors in a dynamic network. Furthermore, they improve the time required to identify an element on the screen. However, they are limited in terms of the accuracy of the visual search. This is because visual stability is achieved by placing multiple actors in the same position at different points in time. It is possible to introduce restrictions to prevent such situations. As a result, the dynamic graph drawing gains a moderate level of visual stability which supports an efficient visual search and a satisfying user experience.

The content of this chapter is based on the publication *"Using Visual Stability to Support Search Efficiency and User Experience in Dynamic Graph Drawings"* [101].

Hypothesis

As it has been reported in Chapter 5, the visual stability of a dynamic graph drawing affects the efficiency of the visual search and the user experience when tracking actors in a dynamic network. On the one hand, those drawings that assign a single actor to a fixed position on the canvas present low levels of visual stability. This results in an unsatisfying user experience with a highly accurate visual search. On the other hand, those drawings that assign multiple actors to a fixed position on the canvas present high levels of visual stability. As a result, a satisfying user experience is obtained but it comes at the cost of losing accuracy on the visual search.

Based on these findings, the research hypothesis for the case study implied that *dynamic graph drawings with a moderate level of visual stability support an efficient visual search and a satisfying user experience when tracking actors in a dynamic network.*

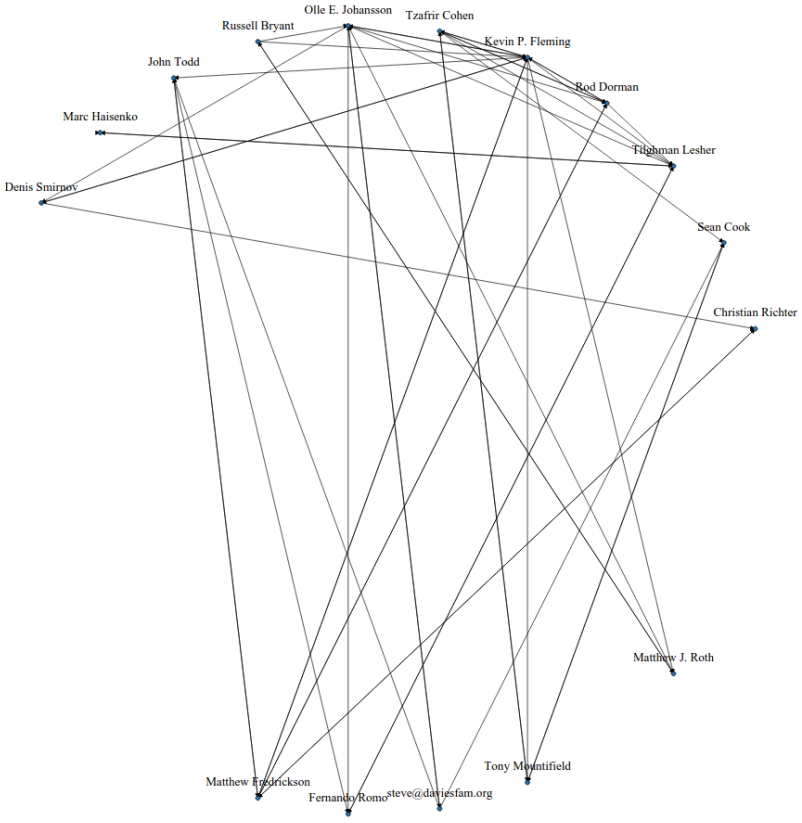

Figure 7.1.: Dynamic graph drawing produced by the two variants of the flickering reduction algorithm for snapshot 1

7.2. Study setup

The setup of the case study is similar to the one described in Chapter 5. It had three scenarios. Each one of them, illustrated the reduced dynamic network from the developer mailing list of the Asterisk framework. The drawings were created with the flickering reduction algorithm and with the degree stabilization algorithm. *Scenario 1* used the flickering reduction algorithm in combination with the flickering index (FRA-FI), *Scenario 2* used the flickering reduction algorithm but with the gap index (FRA-GI), while *Scenario 3* used the degree stabilization algorithm (DSA). The resulting drawings are shown in Figures 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8, 7.9 and 7.10 respectively. The order in which the scenarios were presented to the participants of the study was assigned randomly.

The participants of the case study were a group of 20 students aged between 22 and 28. They were from the fields of computer science and applied cognitive science. All students had a basic knowledge regarding social network analysis techniques. However, they were not familiar with the theories of visual stability nor the mental map. The students were requested to complete the same tasks presented in Chapter 5. There was no time limit for completing the tasks or the scenarios. Still, the average duration per students for all scenarios was 30 minutes.

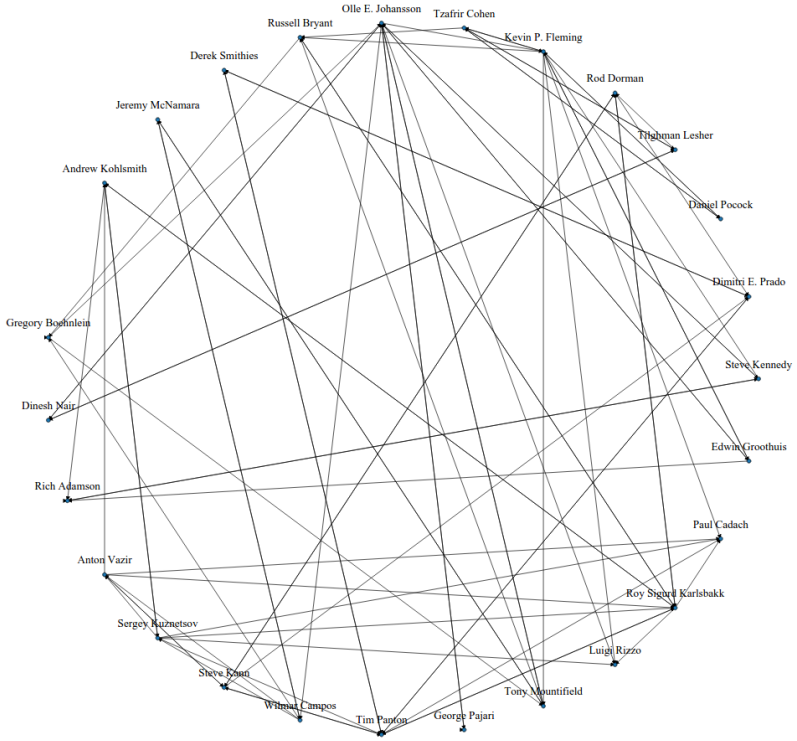

Figure 7.2.: Dynamic graph drawing produced by the two variants of the flickering reduction algorithm for snapshot 2

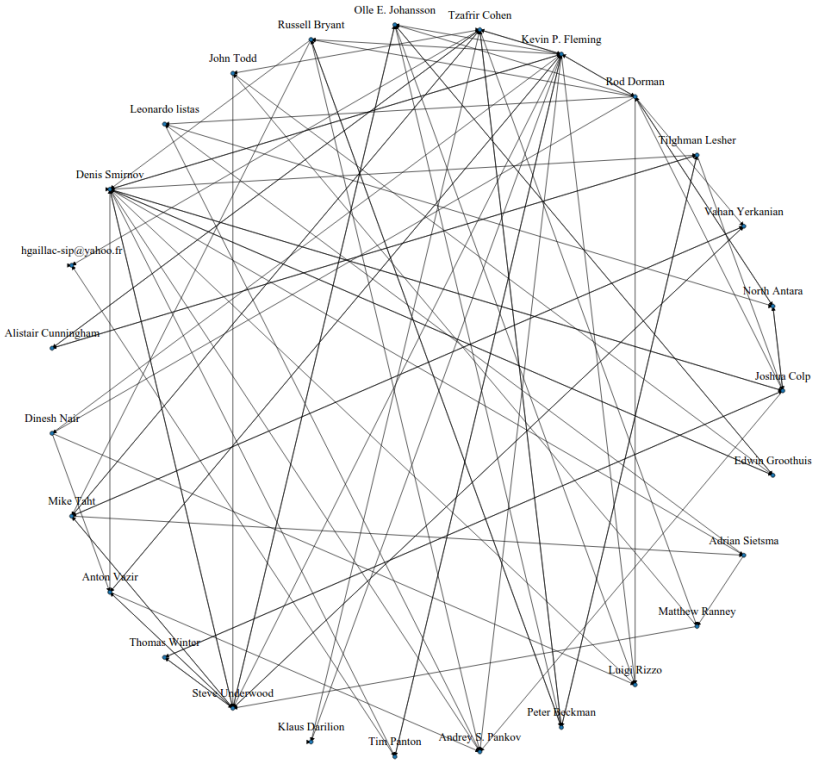

Figure 7.3.: Dynamic graph drawing produced by the two variants of the flickering reduction algorithm for snapshot 3

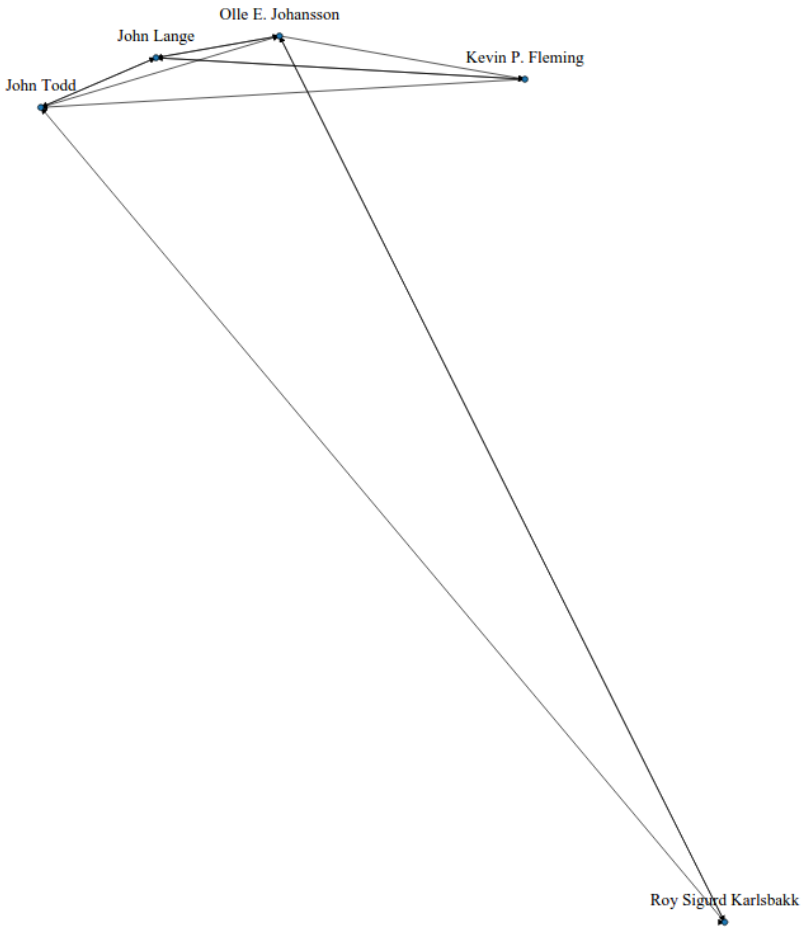

Figure 7.4.: Dynamic graph drawing produced by the two variants of the flickering reduction algorithm for snapshot 4

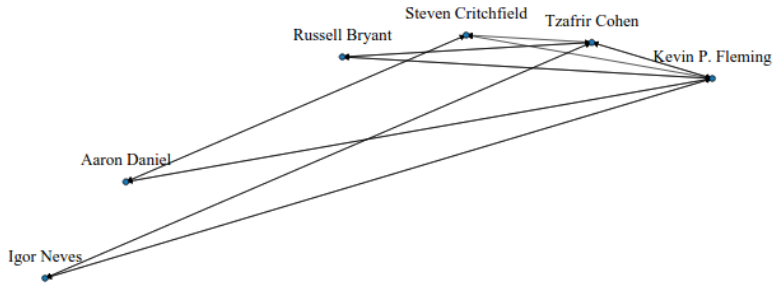

Figure 7.5.: Dynamic graph drawing produced by the two variants of the flickering reduction algorithm for snapshot 5

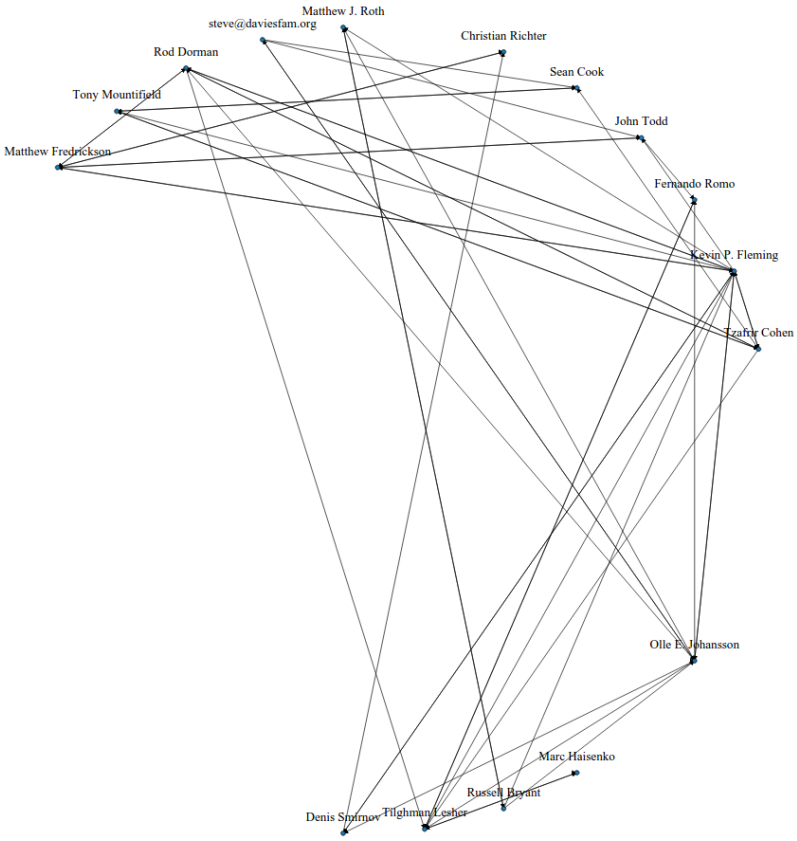

Figure 7.6.: Dynamic graph drawing produced by degree stabilization algorithm for snapshot 1

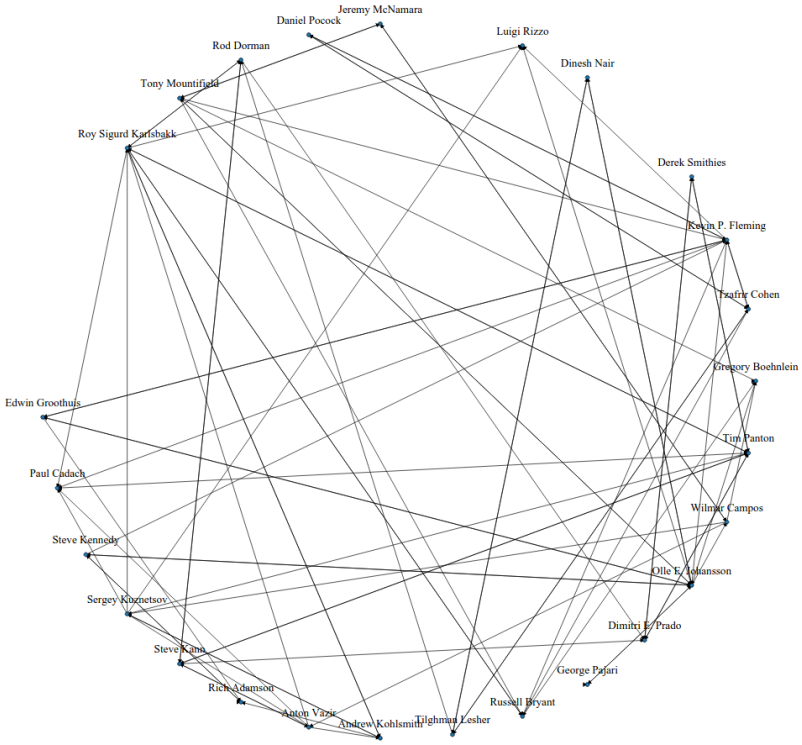

Figure 7.7.: Dynamic graph drawing produced by degree stabilization algorithm snapshot 2

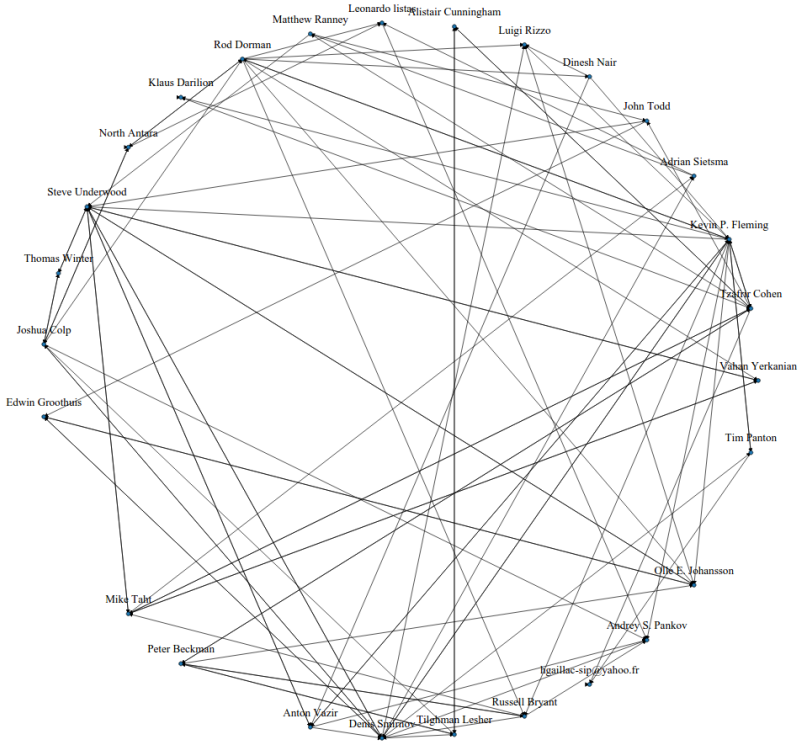

Figure 7.8.: Dynamic graph drawing produced by degree stabilization algorithm snapshot 3

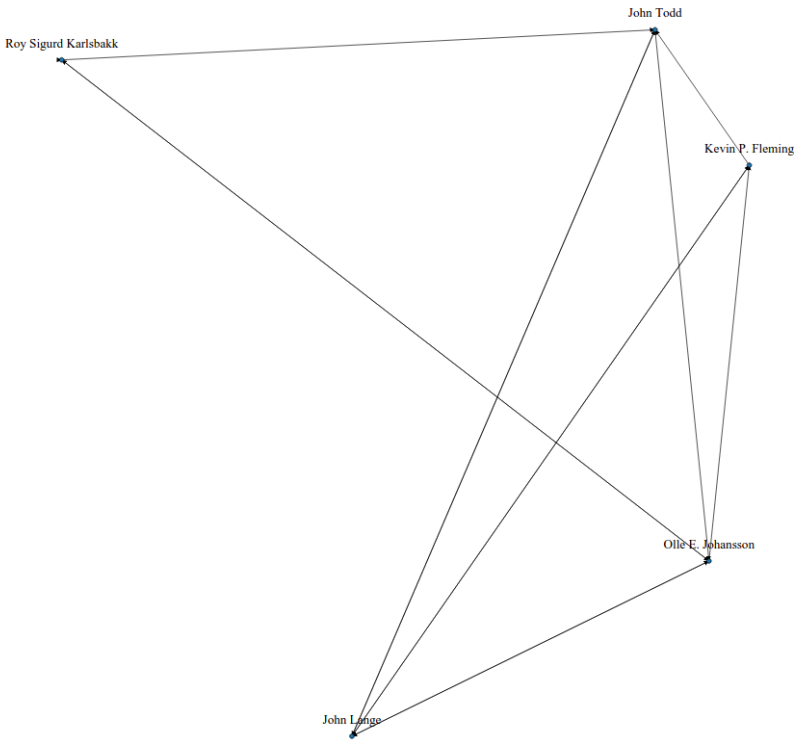

Figure 7.9.: Dynamic graph drawing produced by degree stabilization algorithm snapshot 4

7.3. Analysis and results

The data collected from the model-based measures, the questionnaires and the eye-tracking device was analyzed following the same procedure as described in Chapter 5.

Model-based metrics

As illustrated in Table 7.1, *FRA-FI* along with the *FRA-GI*, had the lowest values in terms of the active positions. The approaches presented 41% (0.412) of the vertex positions as active and 14% (0.141) of the edge positions (EDAP) in the same state. In contrast, the *DSA* had 42% (0.421) of the vertex positions as active, while 13% (0.130) of the edge positions were in the same state. All the drawings assign a fixed position to each actor in the Euclidean Space. Therefore, the graph drawing offset (GDO) was zero.

The other model-based metrics presented more significant results. All dynamic graph drawings obtained a vertex set stability (VS) of 16% (0.165), an edge set stability (ES) of 1% (0.015) and a vertex set degree change (VDC) of 41% (0.413). However, the metrics related to the stability of the drawing showed something different. The *FRA-FI* along with the *FRA-GI* obtained a vertex set drawing stability (VDS) of 53% (0.535), an edge set drawing stability (EDS) of 6% (0.064) and a vertex set neighborhood change (VDNC) of 39% (0.392). In the case of the *DSA*, it obtained a vertex set drawing stability (VDS)

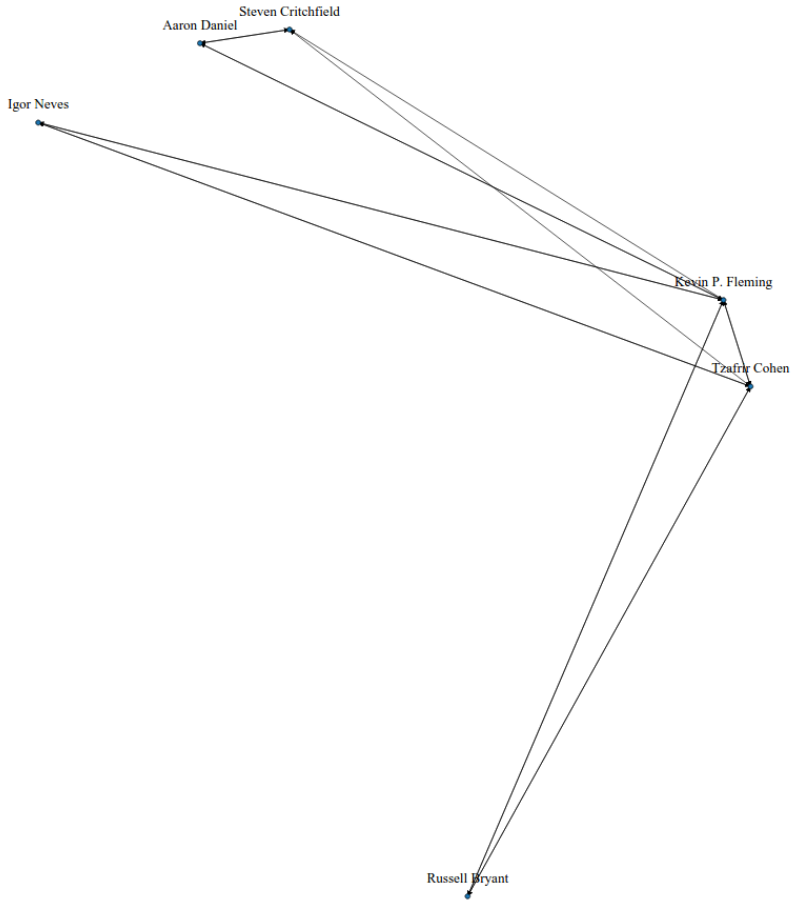

Figure 7.10.: Dynamic graph drawing produced by degree stabilization algorithm snapshot 5

Table 7.1.: Average visual stability calculated for the three drawings under study

Dimensions of the model	FRA-FI	FRA-GI	DSA
VDAP vertex set drawing active positions	0.412	0.412	0.421
EDAP edge set drawing active positions	0.141	0.141	0.130
GDO graph drawing offset	0.000	0.000	0.000
VS vertex set stability	0.165	0.165	0.165
VDS vertex set drawing stability	0.535	0.535	0.388
ES edge set stability	0.015	0.015	0.015
EDS edge set drawing stability	0.064	0.064	0.044
VDC vertex set degree change	0.413	0.413	0.413
VDNC vertex set drawing neighborhood change	0.392	0.392	0.388

of 38% (0.388), an edge set drawing stability (EDS) of 4% (0.044) and a neighborhood change (VDNC) of 38% (0.388).

The results obtained from the model-based metrics indicate that the visual stability of the *FRA-FI* and the *FRA-GI* is even higher than the one reported for the foresighted circular layout (FCL) in Chapter 5. On the contrary, the visual stability of the *DSA* was found to be lower than the one reported for the *FCL*, *FRA-FI* and *FRA-GI* but higher than circular super graph (CSG). In other words, the dynamic graph drawing produced by the *DSA* has a moderate level of visual stability.

Questionnaires

Scenario 1 presented the dynamic social network with the *FRA-FI*, obtaining *10:38* minutes as an average execution time for all the tasks. As illustrated in Figure 7.11, the *FRA-FI* was considered to be of use for tracking actors in a dynamic social network (C2 = 3.400). The students noticed that the actors were assigned to a fixed position in the Euclidean Space (C1 = 4.150), but had some problems to locate the requested actors (C4 = 3.150). This was the case because the users were confused to observe multiple actors in the same position at different points in time. Despite this fact, the changes in the dynamic network were not perceived (C3 = 2.850), while the addition and removal of elements from the drawing were not distracting (C5 = 2.500).

With regard to the open questions, four categories were found for the *FRA-FI*. *Usability strengths* reported that the drawings were useful to track actors over time since it maintained their position in the Euclidean Space. *Drawing overview* reported that the dynamic graph drawing always maintained a constant shape. Such a characteristic was

found as positive by the users. *Location* reported that it was easy to locate a given actor and thus was considered as positive. Nevertheless, the *limitations* category highlighted several negative aspects of the *FRA-FI*. It reported that once the desired actor has been found, it is easy to track him over time. Yet, observing multiple actors in the same position is confusing. Lastly, the drawing was considered not to be intuitive and lacked features to locate faster the desired actor.

Scenario 2 presented the dynamic social network with the *FRA-GI*, obtaining *09:06* minutes as an average execution time for all the tasks. As shown in Figure 7.11, the *FRA-GI* was considered to be less useful in comparison to the *FRA-FI* (C2 = 3.300). The students noticed that the actors were assigned to a fixed position in the Euclidean Space (C1 = 4.250). However, they had problems locating the requested actors (C4 = 3.250) because multiple actors were displayed in the same position during the temporal navigation. Nonetheless, the changes in the dynamic graph drawing were not perceived (C3 = 2.950). Likewise, the addition and removal of elements were not distracting (C5 = 2.200).

In terms of the open questions, four categories were found for the *FRA-GI*. *Location* reported that it was easy to locate an actor in the dynamic network; an aspect which was considered positive by the users. *Drawing overview* reported that the users liked positive the constant shape of the drawing. As *usability strengths*, the users found positive that the drawing maintained the position of the actors in the Euclidean Space. Despite these facts, the *limitations* category pointed out some negative aspects of the of the *FRA-GI*. It was reported that as long as an actor has been found, it is very easy to track him. Still, it is confusing to observe multiple actors in the same position at different points in time. This made the users feel insecure about the location of the actor during the temporal navigation. Finally, the drawing did not reflect the changes occurring in the dynamic network.

Scenario 3 presented the dynamic social network with the *DSA* obtaining *8:55* minutes as an average execution time for all the tasks. As illustrated in Figure 7.11, the *DSA* was considered to be less useful in comparison to the previous approaches (C2 = 3.200). The students noticed that the actors were maintaining their position in the Euclidean Space (C1 = 3.800) and the students had fewer problems to locate the requested actors (C4 = 2.900). Nonetheless, the changes in the dynamic graph drawing were perceived to a certain point (C3 = 3.100), but the addition and removal of elements from the drawing were not distracting (C5 = 2.450).

With regard to the open questions, four categories were found for the *DSA*. *Location* reported that it was easy to locate an actor in the dynamic network. This aspect was considered positive by the users. *Usability strengths* reported that the drawing always maintained the position of the actors in the Euclidean Space and also that it was easy to explore the dynamic graph drawing. Both aspects were considered positives. *Drawing overview* reported that the constant shape of the dynamic graph drawing was perceived as positive. However, the *limitations* category exposed some negative aspects of the *DSA*. It was reported that only when an actor has been found, it is easy to track him over time. Yet, it is still confusing to observe multiple actors in the same position but not as much as in the other approaches. Lastly, the dynamic graph drawing was considered to be useful for tracking actors but not relation related patterns.

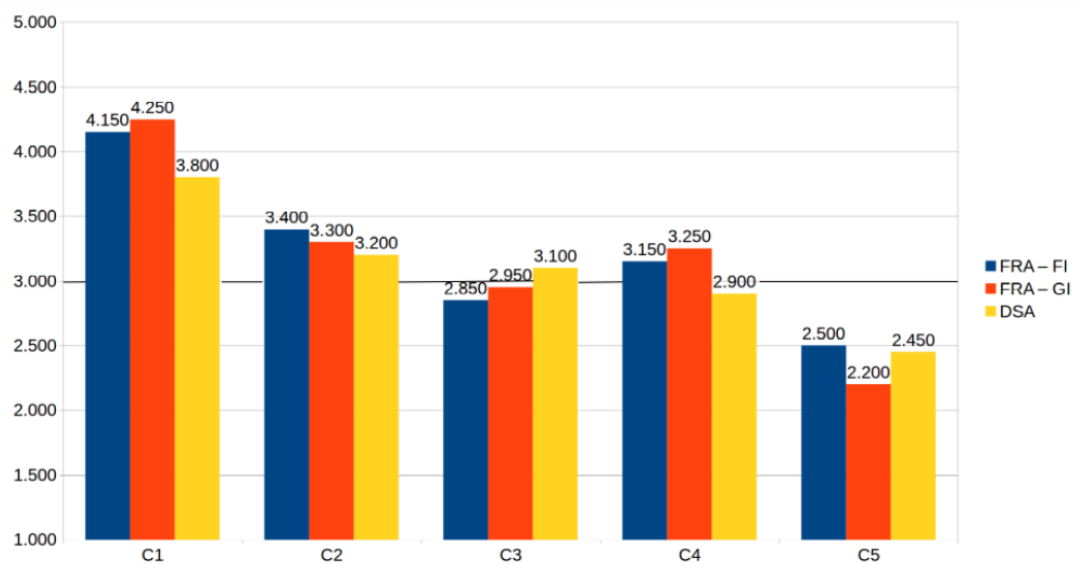

Figure 7.11.: Results of the questionnaires for the five criteria

Eye-tracking metrics on the overall drawing area

Number of fixations

As shown in Table 7.2, the *FRA-GI* had the best performance regarding the number of fixations. It obtained an average of 57.570 fixations, followed by the *DSA* with 65.210 and the *FRA-FI* with 69.690. According to this result, the students executed fewer fixations using the *FRA-GI* in comparison to the other approaches. Nevertheless, the difference between the approaches was found not to be statistically significant as illustrated in Table 5.4.

Duration of fixations

Table 7.4 depicts that, the *FRA-GI* had the best performance regarding the duration of the fixations. It obtained an average of 0.196 seconds, followed by the *DSA* with 0.197 and the *FRA-FI* with 0.208. According to this result, the students required more time to recognize an element using the *FRA-GI* in comparison to the other approaches. Nonetheless, the difference was found not to be statistically significant as illustrated in Table 7.5.

Number of saccades

As shown in Table 7.6, the *FRA-GI* had the best performance regarding the number of saccades. It obtained an average of 58.250 saccades, followed by the *DSA* with 66.190 and the *FRA-FI* with 70.730. According to the results, the students executed fewer saccades using the *FRA-GI*. Nevertheless, the difference between the evaluated approaches was found not to be statistically significant as displayed in Table 7.7.

Table 7.2.: Number of fixations on the overall drawing area

Number of Fixations			
ID	**FRA-FI**	**FRA-GI**	**DSA**
1163866620	64.800	81.200	86.400
1169451400	72.800	95.000	104.400
3341759460	93.600	85.600	75.200
4291681437	68.000	32.400	66.400
4295024649	73.600	39.600	42.400
4328697256	75.600	45,200	46,200
4407069149	57.600	40.400	42.800
4999902580	44.800	23.400	39.000
5032930026	67.600	62.600	56.000
5308616338	54.600	38.000	70.000
6385045328	87.600	65.200	88.800
6536263994	84.000	75.600	96.000
670690418	39.600	30.400	56.400
698441774	48.800	68.800	43.000
7026726368	74.400	22.000	96.800
7344799308	59.400	76.800	46.800
7507269062	60.000	57.200	89.600
7814128462	118.200	83.400	96.000
8969515821	20.800	88.000	24.800
9849432742	128.000	40.600	37.200
Average	**69.690**	**57.570**	**65.210**
Standard Deviation	**25.015**	**23.496**	**24.749**

Table 7.3.: T-test for the number of fixations on the overall drawing area

	T-test		
	FRA-FI - FRA-GI	**DSA - FRA-FI**	**DSA - FRA-GI**
Z	-1.829b	-0.336b	-1.493c
p	0.067	0.737	0.135

Table 7.4.: Duration of fixations in seconds on the overall drawing area

Duration of Fixations			
ID	**FRA-FI**	**FRA-GI**	**DSA**
1163866620	0.316	0.311	0.287
1169451400	0.316	0.291	0.287
3341759460	0.314	0.178	0.202
4291681437	0.285	0.201	0.237
4295024649	0.212	0.233	0.239
4328697256	0.145	0.117	0.257
4407069149	0.256	0.237	0.222
4999902580	0.163	0.162	0.131
5032930026	0.279	0.246	0.255
5308616338	0.214	0.177	0.222
6385045328	0.162	0.181	0.162
6536263994	0.296	0.352	0.265
670690418	0.129	0.109	0.131
698441774	0.122	0.133	0.124
7026726368	0.129	0.099	0.166
7344799308	0.124	0.160	0.202
7507269062	0.251	0.229	0.168
7814128462	0.170	0.221	0.149
8969515821	0.110	0.162	0.131
9849432742	0.165	0.126	0.108
Average	**0.208**	**0.196**	**0.197**
Standard Deviation	**0.074**	**0.069**	**0.058**

Table 7.5.: T-test for the duration of fixations on the overall drawing area

	T-test		
	FRA-FI - FRA-GI	**DSA - FRA-FI**	**DSA - FRA-GI**
Z	-.971b	-1.195b	-0.112b
p	0.332	0.232	0.911

Table 7.6.: Number of saccades on the overall drawing area

Number of Saccades			
ID	**FRA-FI**	**FRA-GI**	**DSA**
1163866620	63.600	79.000	84.400
1169451400	72.600	95.000	103.600
3341759460	92.400	85.800	76.000
4291681437	67.600	31.400	64.400
4295024649	74.400	38.400	41.600
4328697256	78.000	46.400	46.400
4407069149	56.200	41.800	44.000
4999902580	47.800	20.800	44.600
5032930026	67.200	61.800	58.200
5308616338	54.200	36.800	68.600
6385045328	89.800	67.200	91.800
6536263994	83.600	76.000	96.200
670690418	42.200	33.200	57.200
698441774	48.400	64.800	46.600
7026726368	77.400	27.400	97.800
7344799308	60.200	77.400	46.200
7507269062	58.800	56.600	92.000
7814128462	124.600	87.200	96.800
8969515821	22.800	88.600	27.000
9849432742	132.800	49.400	40.400
Average	**70.730**	**58.250**	**66.190**
Standard Deviation	**25.970**	**23.131**	**24.148**

Table 7.7.: T-test for the number of saccades on the overall drawing area

	T-test		
	FRA-FI - FRA-GI	**DSA - FRA-FI**	**DSA - FRA-GI**
Z	-1.904b	-0.261b	-1.489c
p	0.057	0.794	0.136

Table 7.8.: Duration of saccades in seconds on the overall drawing area

Duration of Saccades			
ID	**FRA-FI**	**FRA-GI**	**DSA**
1163866620	0.019	0.015	0.020
1169451400	0.020	0.020	0.024
3341759460	0.029	0.047	0.032
4291681437	0.020	0.024	0.023
4295024649	0.032	0.025	0.029
4328697256	0.134	0.330	0.024
4407069149	0.036	0.042	0.060
4999902580	0.120	0.195	0.180
5032930026	0.021	0.038	0.032
5308616338	0.038	0.019	0.024
6385045328	0.158	0.075	0.060
6536263994	0.034	0.025	0.045
670690418	0.220	0.195	0.166
698441774	0.082	0.204	0.486
7026726368	0.237	1.312	0.062
7344799308	0.187	0.104	0.022
7507269062	0.025	0.033	0.106
7814128462	0.099	0.073	0.072
8969515821	0.785	0.055	0.338
9849432742	0.076	0.307	0.322
Average	**0.119**	**0.157**	**0.106**
Standard Deviation	**0.172**	**0.289**	**0.130**

Duration of saccades

Table 7.8 points out that the *DSA* had the best performance regarding the duration of the saccades. It obtained an average of 0.106 seconds, followed by the *FRA-FI* with 0.119 seconds and the *FRA-GI* with 0.157 seconds. According to these results, the students executed faster eye movements when using the *DSA*. However, the difference between the evaluated approaches was found not to be statistically significant as depicted in Table 7.9.

Table 7.9.: T-test for the duration of saccades on the overall drawing area

	T-test		
	FRA-FI - FRA-GI	**DSA - FRA-FI**	**DSA - FRA-GI**
Z	-0.336b	-0.373c	-0.112c
p	0.737	0.709	0.911

Table 7.10.: Saccadic amplitude in pixels on the overall drawing area

Saccadic Amplitude			
ID	**FRA-FI**	**FRA-GI**	**DSA**
1163866620	97.836	79.593	104.967
1169451400	118.052	103.034	147.658
3341759460	129.462	129.942	139.552
4291681437	137.617	171.360	168.504
4295024649	130.398	153.342	144.211
4328697256	129.756	189.117	122.659
4407069149	123.764	149.415	164.251
4999902580	133.810	341.345	202.720
5032930026	121.983	131.379	146.410
5308616338	124.499	109.407	156.295
6385045328	148.919	155.867	143.793
6536263994	96.043	100.089	131.674
670690418	143.739	207.715	205.463
698441774	132.693	271.619	186.522
7026726368	124.126	303.084	160.730
7344799308	187.837	131.563	141.847
7507269062	156.038	138.522	144.641
7814128462	136.226	149.953	137.270
8969515821	230.660	103.536	199.426
9849432742	127.876	240.366	206.498
Average	**136.567**	**168.012**	**157.755**
Standard Deviation	**29.435**	**71.289**	**28.919**

Saccadic amplitude

As emphasized in Table 7.10, the *FRA-FI* had the best performance regarding the saccadic amplitude. The approach presented an average distance of 136.567 pixels, followed by the *DSA* with 157.755 pixels and the *FRA-GI* with 168.012 pixels. According to this result, the eye movements executed by the students covered less distance using the *FRA-FI*. This was found to be statistically significant in comparison to the *DSA* ($[Z = -2.501b, p = 0.012]$). Nonetheless, there was no evidence suggesting that the *FRA-FI* was superior than the *FRA-GI* or vice versa, as illustrated in Table 7.11.

Table 7.11.: T-test for the saccadic amplitude on the overall drawing area

	T-test		
	FRA-FI - FRA-GI	**DSA - FRA-FI**	**DSA - FRA-GI**
Z	-1.680b	-2.501b	-0.075c
p	0.093	0.012	0.940

Table 7.12.: Scan path length in pixels on the overall drawing area

Scan Path Length			
ID	**FRA-FI**	**FRA-GI**	**DSA**
1163866620	2451.329	3461.332	3243.827
1169451400	2451.329	3461.332	3243.827
3341759460	2451.329	3461.332	3243.827
4291681437	2341.782	2427.094	2035.924
4295024649	3111.692	2110.235	3514.086
4328697256	2818.711	2661.198	2529.802
4407069149	2463.579	3347.224	3373.064
4999902580	1053.046	480.183	1161.246
5032930026	2445.606	3798.975	1649.555
5308616338	3083.604	2706.223	5584.941
6385045328	3532.366	3167.312	3699.943
6536263994	3974.781	3243.257	3268.803
670690418	2375.443	2236.701	4449.933
698441774	3324.151	2653.839	1475.574
7026726368	1388.349	468.909	6082.903
7344799308	5879.353	6519.781	4314.142
7507269062	5410.760	4223.924	3773.460
7814128462	1703.817	1853.096	4938.735
8969515821	2384.262	4616.045	1108.061
9849432742	4158.913	1895.461	2668.768
Average	**3223.198**	**3137.066**	**3652.300**
Standard Deviation	**1457.535**	**1587.170**	**1843.828**

Scan path length

As displayed in Table 7.12, the *FRA-GI* had the best performance regarding the scan path length. It obtained an average of 3137.066 pixels, followed by the *FRA-FI* with 3223.198 pixels and the *DSA* with 3652.300 pixels. According to this result, the visual search covered the shortest distance using the *FRA-GI*. Still, the difference between the evaluated approaches was found not to be statistically significant as shown in Table 7.13.

Table 7.13.: T-test for the scan path length on the overall drawing area

	T-test		
	FRA-FI - FRA-GI	**DSA - FRA-FI**	**DSA - FRA-GI**
Z	-0.560b	-0.411c	-0.859c
p	0.575	0.681	0.391

Table 7.14.: Scan Path Duration in seconds on the overall drawing area

Scan Path Duration			
ID	**FRA-FI**	**FRA-GI**	**DSA**
1163866620	6.710	8.981	7.811
1169451400	9.166	9.543	15.099
3341759460	15.162	11.042	9.641
4291681437	4.569	3.394	3.589
4295024649	5.723	3.340	6.238
4328697256	7.574	10.044	4.564
4407069149	4.855	5.409	5.455
4999902580	3.199	1.868	3.424
5032930026	4.739	7.691	3.258
5308616338	5.493	3.765	7.392
6385045328	7.675	5.464	6.671
6536263994	10.693	9.187	6.993
670690418	7.629	4.910	8.810
698441774	6.231	7.239	6.821
7026726368	5.995	2.234	8.844
7344799308	10.732	11.263	6.137
7507269062	8.708	6.237	8.590
7814128462	4.928	4.723	8.872
8969515821	9.336	8.967	4.175
9849432742	8.844	5.673	9.181
Average	**7.398**	**6.549**	**7.078**
Standard Deviation	**2.797**	**2.920**	**2.781**

Scan path duration

Table 7.14 shows that the *FRA-GI* had the best performance regarding the scan path duration. It obtained an average of 7.398 seconds, followed by the *DSA* with 7.078 and the *FRA-FI* with 7.398. According to this result, the fastest visual search is provided by the *FRA-GI*. Still, the difference was found not to be statistically significant as pointed in Table 7.15.

Spatial density

As indicated in Table 7.16, the *FRA-GI* had the best results in terms of the spatial density. It obtained an average of 0.357, followed by the *FRA-FI* with 0.374 and the

Table 7.15.: T-test for the scan path duration on the overall drawing area

	T-test		
	FRA-FI - FRA-GI	**DSA - FRA-FI**	**DSA - FRA-GI**
Z	-1.643b	-0.187b	-0.672c
p	0.1	0.852	0.502

Table 7.16.: Spatial density on the overall drawing area

Spatial Density			
ID	**FRA-FI**	**FRA-GI**	**DSA**
1163866620	0.370	0.396	0.406
1169451400	0.360	0.348	0.438
3341759460	0.378	0.472	0.508
4291681437	0.428	0.356	0.520
4295024649	0.418	0.386	0.410
4328697256	0.400	0.360	0.426
4407069149	0.330	0.332	0.428
4999902580	0.348	0.196	0.280
5032930026	0.386	0.380	0.438
5308616338	0.370	0.256	0.468
6385045328	0.386	0.354	0.532
6536263994	0.444	0.426	0.478
670690418	0.330	0.352	0.326
698441774	0.324	0.368	0.358
7026726368	0.366	0.210	0.494
7344799308	0.410	0.430	0.422
7507269062	0.444	0.404	0.460
7814128462	0.324	0.378	0.468
8969515821	0.240	0.408	0.172
9849432742	0.428	0.328	0.300
Average	**0.374**	**0.357**	**0.417**
Standard Deviation	**0.050**	**0.069**	**0.090**

DSA with 0.417. According to this, the *FRA-GI* provides the most direct visual search. This was found to be statistically significant in comparison to the *DSA* ($[Z = 2.671c, p = 0.008]$). Likewise, the visual search was more direct using the *FRA-FI* in comparison to the *DSA* ($[Z = 2.336c, p = 0.019]$). Nevertheless, there was no evidence suggesting that the *FRA-FI* was superior to the *FRA-GI* or vice versa.

Fixation/saccade ratio

As shown in Table 7.18, the *DSA* had the best performance in terms of the fixation/saccade ratio. It obtained an average ratio of 5.965, followed by the *FRA-GI* with 6.211 and the *FRA-FI* with 6.803. According to these results, the *DSA* provided the

Table 7.17.: T-test for the spatial density on the overall drawing area

	T-test		
	FRA-FI - FRA-GI	**DSA - FRA-FI**	**DSA - FRA-GI**
Z	-0.947b	-2.336c	-2.671c
p	0.344	0.019	0.008

Table 7.18.: Fixation/saccade ratio on the overall drawing area

Fixation/Saccade Ratio			
ID	**FRA-FI**	**FRA-GI**	**DSA**
1163866620	17.448	21.959	14.691
1169451400	17.409	14.600	13.036
3341759460	11.669	4.280	6.958
4291681437	15.756	9.091	11.415
4295024649	7.959	10.318	8.737
4328697256	1.666	0.391	12.133
4407069149	10.469	7.108	6.323
4999902580	1.445	0.954	0.777
5032930026	13.625	7.213	9.001
5308616338	6.907	7.597	9.472
6385045328	1.666	3.394	3.295
6536263994	10.309	15.368	6.542
670690418	1.103	0.649	1.140
698441774	1.571	0.767	0.373
7026726368	0.602	0.267	3.057
7344799308	0.873	2.748	7.303
7507269062	10.385	9.616	2.064
7814128462	1.839	3.236	2.132
8969515821	0.204	4.174	0.461
9849432742	3.152	0.489	0.394
Average	**6.803**	**6.211**	**5.965**
Standard Deviation	**6.124**	**5.936**	**4.671**

visual search with the longest distance as it is confirmed by the scan path length. Nevertheless, the difference between the evaluated approaches was found not to be statistically significant as illustrated in 7.19.

Eye-tracking metrics on the areas of interest

The AOIs were placed in the Euclidean Space according to the positioning rules of the dynamic graph drawing algorithms. The *FRA-FI*, the *FRA-GI* and the *DSA* assigned the actors of the dynamic network a fixed position in the Euclidean Space. Therefore, it was not necessary to adjust the AOIs as shown in Figures 7.12, 7.13, 7.14, 7.15, 7.16, 7.17, 7.18, 7.19, 7.20 and 7.21.

Table 7.19.: T-test for the fixation/saccade ratio on the overall drawing area

	T-test		
	FRA-FI - FRA-GI	**DSA - FRA-FI**	**DSA - FRA-GI**
Z	-0.560b	-1.195b	-0.149b
p	0.575	0.232	0.881

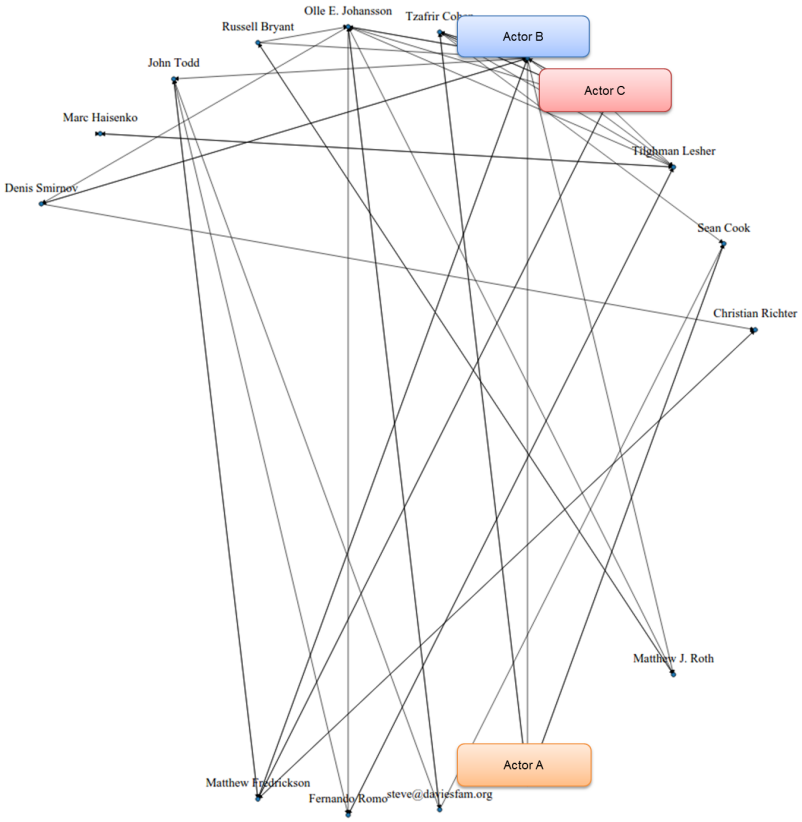

Figure 7.12.: Areas of interest in the dynamic graph drawing produced by both variants of the flickering reduction algorithm for snapshot 1

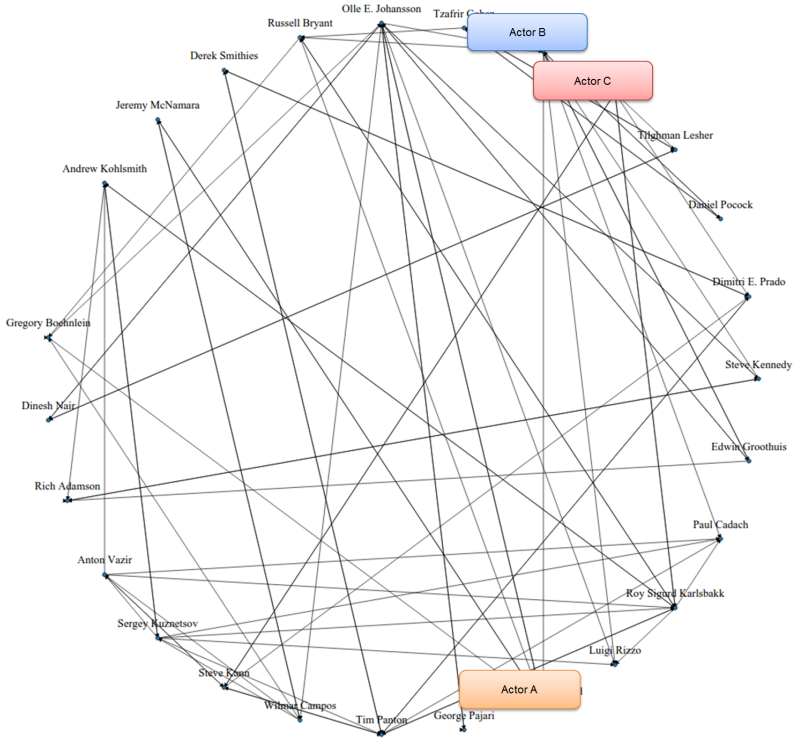

Figure 7.13.: Areas of interest in the dynamic graph drawing produced by both variants of the flickering reduction algorithm for snapshot 2

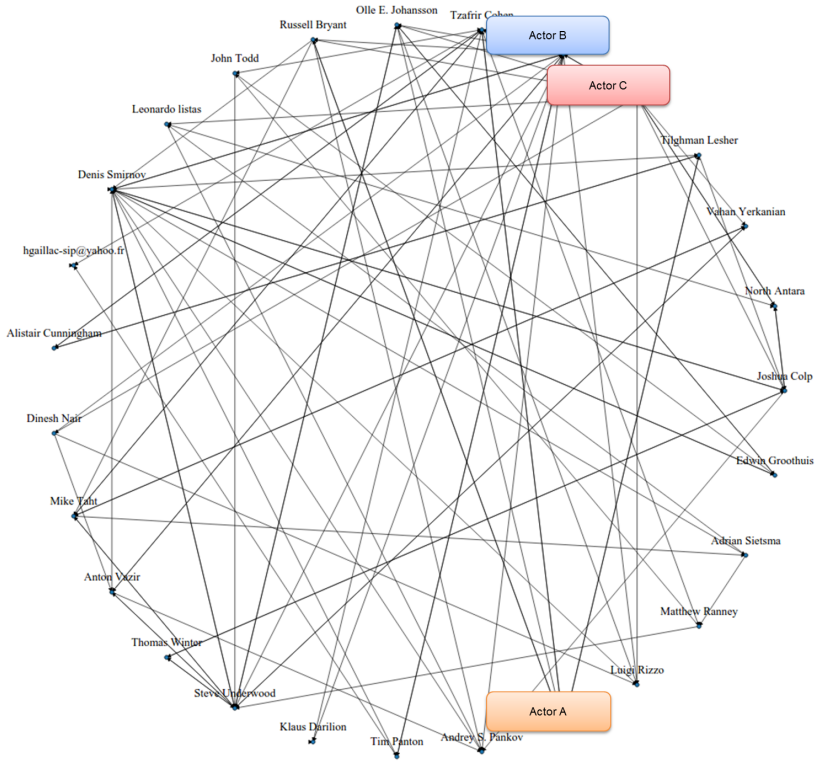

Figure 7.14.: Areas of interest in the dynamic graph drawing produced by both variants of the flickering reduction algorithm for snapshot 3

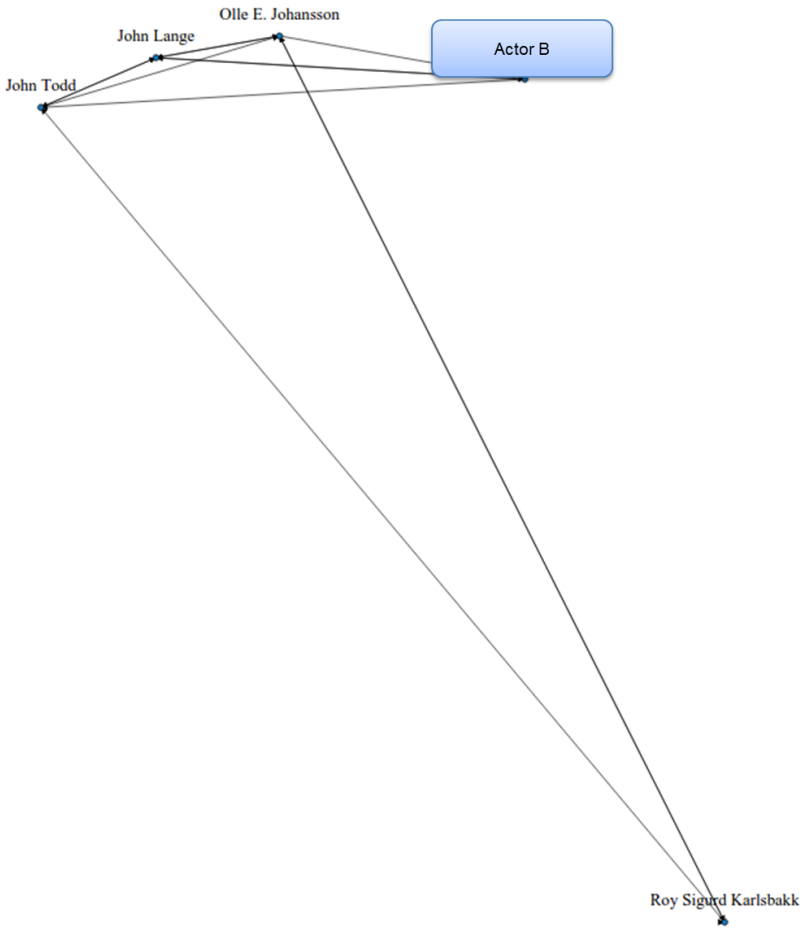

Figure 7.15.: Areas of interest in the dynamic graph drawing produced by both variants of the flickering reduction algorithm for snapshot 4

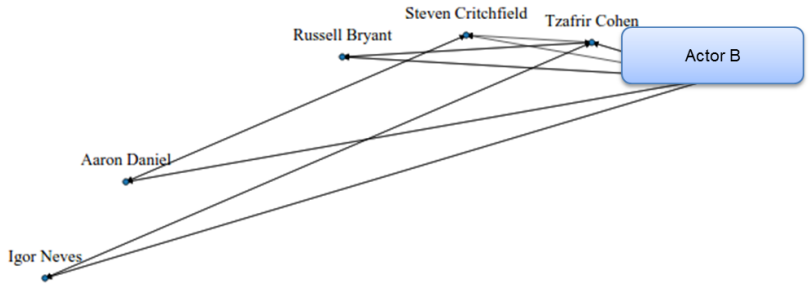

Figure 7.16.: Areas of interest in the dynamic graph drawing produced by both variants of the flickering reduction algorithm for snapshot 5

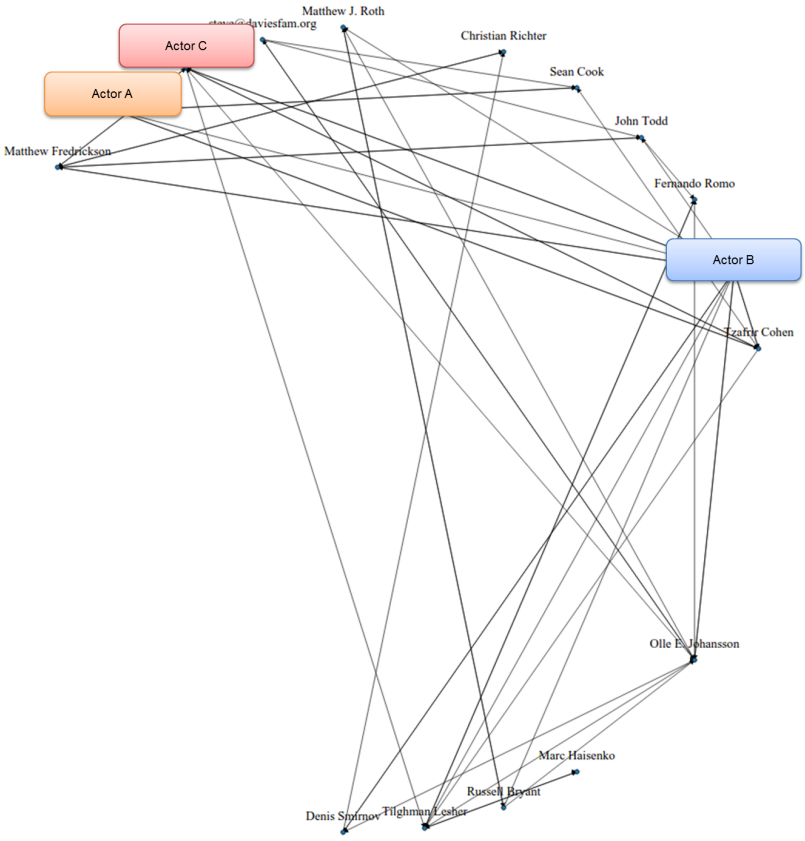

Figure 7.17.: Areas of interest in the dynamic graph drawing produced by the degree stabilization algorithm for snapshot 1

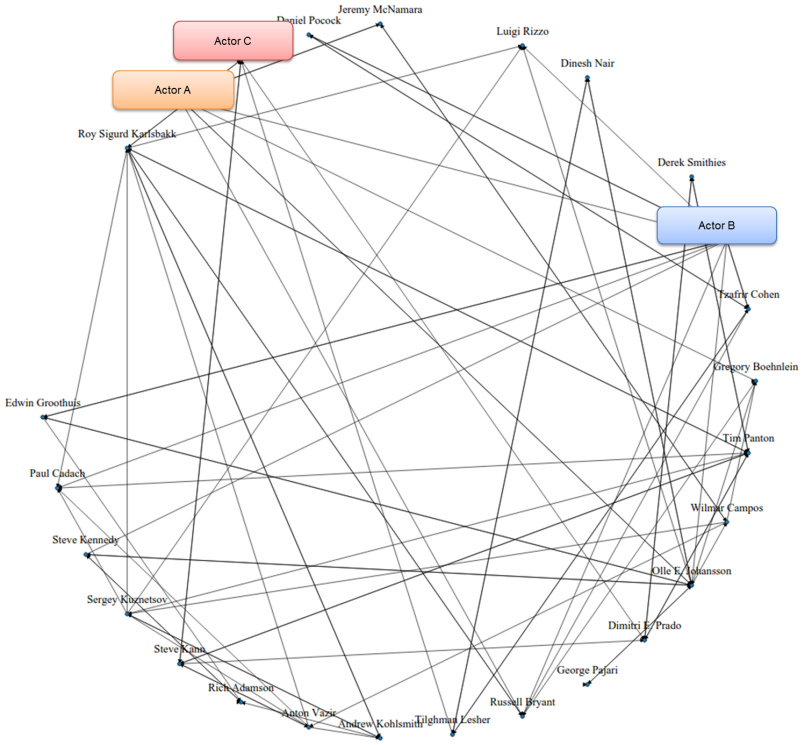

Figure 7.18.: Areas of interest in the dynamic graph drawing produced by the degree stabilization algorithm for snapshot 2

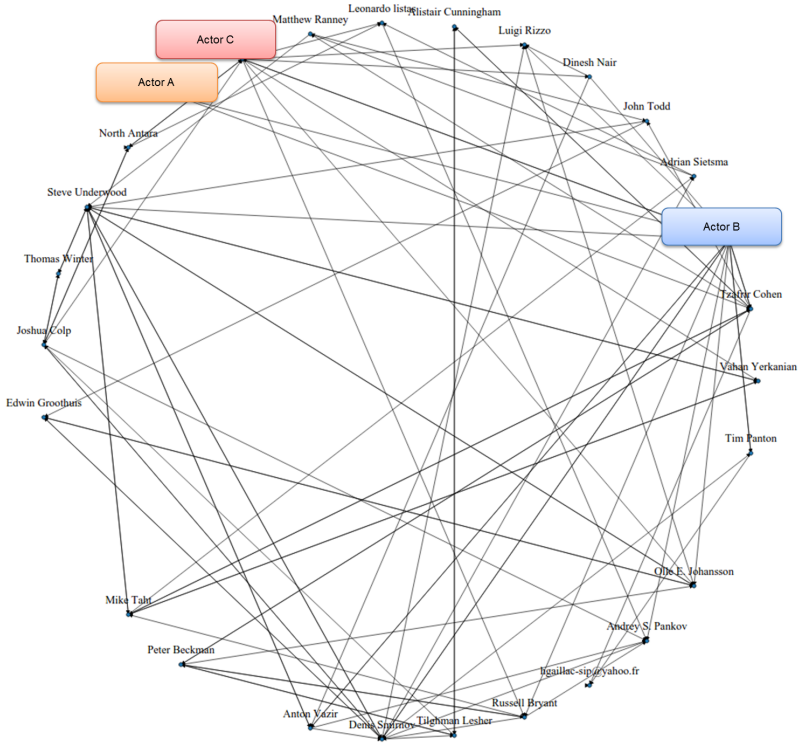

Figure 7.19.: Areas of interest in the dynamic graph drawing produced by the degree stabilization algorithm for snapshot 3

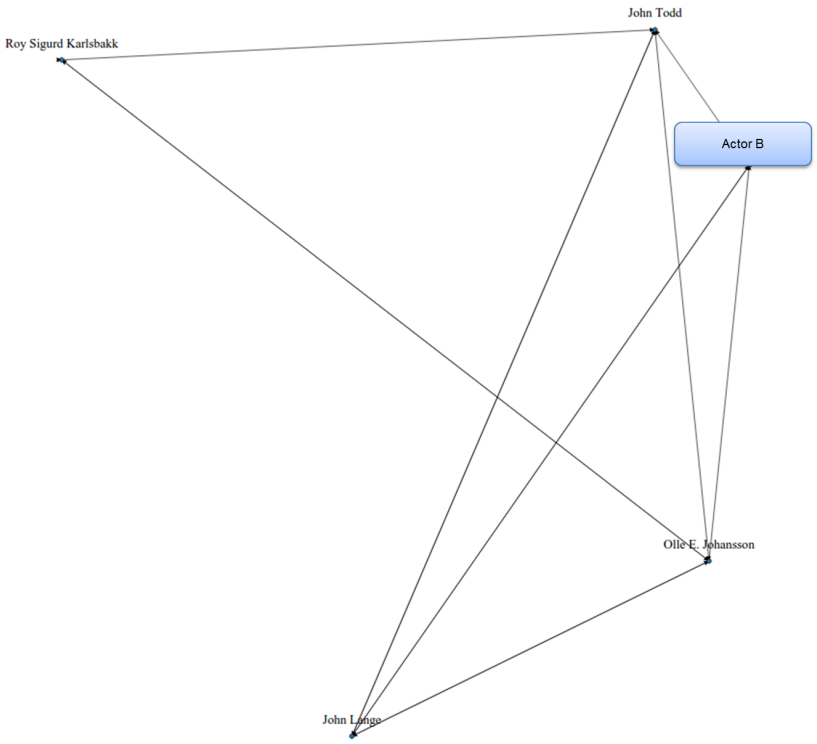

Figure 7.20.: Areas of interest in the dynamic graph drawing produced by the degree stabilization algorithm for snapshot 4

Fixations on target

As shown in Table 7.20, the *DSA* had the best performance in terms of the fixations on target for *Actor A*. It obtained an average of 5.8% (0.058) of the fixations on target, followed by the *FRA-FI* with 3.0% (0.030) and the *FRA-GI* 2.5% (0.025). According to these results, the *DSA* offered the most accurate visual search to locate *Actor A*. This was found to be statistically significant in comparison to the *FRA-FI* ($[Z = -2.792c, p = 0.005]$) and *FRA-GI* ($[Z = -3.210, p = 0.001]$). There was no evidence that the *FRA-FI* was superior to the *FRA-GI* or vice versa, as illustrated in Table 7.21.

As depicted in Table 7.22, the *DSA* had the best performance regarding the fixations on target for *Actor B*. It obtained an average of 8.7% (0.087) of the fixations on target, followed by the *FRA-FI* with 8.2% (0.082) and the *FRA-GI* with 3.7% (0.037). According to this, the *DSA* offered the most accurate visual search for locating *Actor B*. This was found to be statistically significant in comparison to the *FRA-GI* ($[Z = -3.263c, p = 0.001]$). Likewise, the *FRA-FI* was found to be superior than the *FRA-GI* ($[Z = -2.990b, p = 0.003]$). There was no evidence suggesting that the *DSA* was superior to the *FRA-FI* or vice versa, as shown in Table 7.23.

As displayed in Table 7.24, the *DSA* had the best performance in terms of the fixations on target for *Actor C*. It obtained an average of 5.7% (0.057) of the fixations on target, followed by the *FRA-FI* with 4.8% (0.048) and the *FRA-GI* with 2.1% (0.021). According to this result, the *DSA* offered the most accurate visual search for locating *Actor C*. This was found to be statistically significant in comparison to the *FRA-GI*

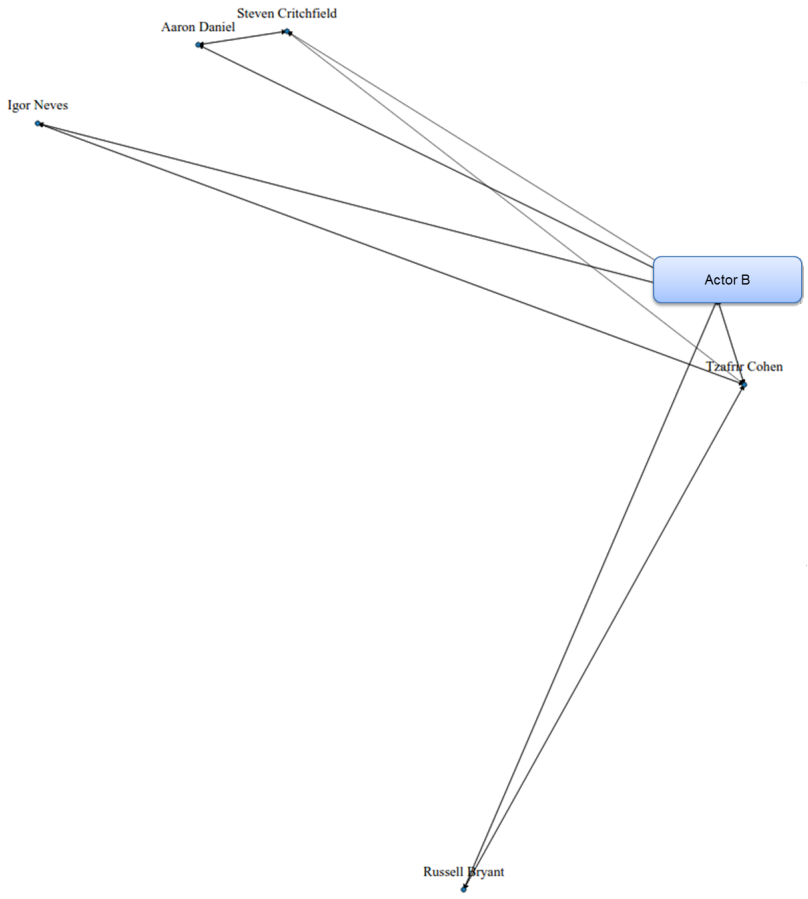

Figure 7.21.: Areas of interest in the dynamic graph drawing produced by the degree stabilization algorithm for snapshot 5

Table 7.20.: Fixations on target for actor A

Fixations on Target - Actor A			
ID	**FRA-FI**	**FRA-GI**	**DSA**
1163866620	0.112	0.012	0.123
1169451400	0.017	0.038	0.077
3341759460	0.027	0.016	0.055
4291681437	0.058	0.069	0.074
4295024649	0.047	0.039	0.010
4328697256	0.054	0.004	0.041
4407069149	0.000	0.062	0.047
4999902580	0.000	0.000	0.012
5032930026	0.000	0.039	0.095
5308616338	0.025	0.041	0.078
6385045328	0.010	0.023	0.049
6536263994	0.041	0.048	0.071
670690418	0.047	0.033	0.000
698441774	0.000	0.013	0.029
7026726368	0.000	0.000	0.067
7344799308	0.011	0.002	0.083
7507269062	0.041	0.004	0.026
7814128462	0.000	0.000	0.051
8969515821	0.033	0.050	0.093
9849432742	0.073	0.000	0.082
Average	**0.030**	**0.025**	**0.058**
Standard Deviation	**0.030**	**0.023**	**0.032**

Table 7.21.: T-test for the fixations on target appearing over actor A

	T-test		
	FRA-FI - FRA-GI	**DSA - FRA-FI**	**DSA - FRA-GI**
Z	-0.215b	-2.792c	-3.210c
p	0.83	0.005	0.001

Table 7.22.: Fixations on target for actor B

Fixations on Target - Actor B			
ID	**FRA-FI**	**FRA-GI**	**DSA**
1163866620	0.015	0.000	0.183
1169451400	0.134	0.071	0.116
3341759460	0.008	0.018	0.089
4291681437	0.129	0.009	0.106
4295024649	0.136	0.074	0.043
4328697256	0.129	0.020	0.104
4407069149	0.046	0.073	0.117
4999902580	0.000	0.000	0.090
5032930026	0.143	0.017	0.057
5308616338	0.156	0.114	0.161
6385045328	0.069	0.009	0.107
6536263994	0.087	0.028	0.168
670690418	0.014	0.073	0.030
698441774	0.130	0.009	0.036
7026726368	0.035	0.043	0.104
7344799308	0.052	0.000	0.011
7507269062	0.055	0.024	0.016
7814128462	0.063	0.005	0.070
8969515821	0.132	0.040	0.018
9849432742	0.108	0.108	0.115
Average	**0.082**	**0.037**	**0.087**
Standard Deviation	**0.052**	**0.036**	**0.051**

Table 7.23.: T-test for the fixations on target appearing over actor B

	T-test		
	FRA-FI - FRA-GI	**DSA - FRA-FI**	**DSA - FRA-GI**
Z	-2.990b	-0.121c	-3.263c
p	0.003	0.904	0.001

Table 7.24.: Fixations on target for actor C

Fixations on Target - Actor C			
ID	**FRA-FI**	**FRA-GI**	**DSA**
1163866620	0.103	0.054	0.104
1169451400	0.070	0.081	0.059
3341759460	0.012	0.000	0.048
4291681437	0.044	0.007	0.068
4295024649	0.048	0.041	0.000
4328697256	0.052	0.008	0.069
4407069149	0.090	0.007	0.087
4999902580	0.000	0.000	0.000
5032930026	0.072	0.000	0.079
5308616338	0.028	0.065	0.112
6385045328	0.033	0.006	0.062
6536263994	0.077	0.016	0.047
670690418	0.019	0.018	0.002
698441774	0.033	0.000	0.018
7026726368	0.037	0.041	0.029
7344799308	0.058	0.007	0.045
7507269062	0.025	0.010	0.096
7814128462	0.036	0.022	0.022
8969515821	0.067	0.024	0.067
9849432742	0.053	0.023	0.116
Average	**0.048**	**0.021**	**0.057**
Standard Deviation	**0.026**	**0.023**	**0.037**

($[Z = -3.124c, p = 0.002]$). Similarly, the *FRA-FI* was more accurate than the *FRA-GI* ($[Z = -3.134b, p = 0.002]$). There was no evidence suggesting that the *DSA* was superior to the *FRA-FI* or vice versa as illustrated in Table 7.25.

Time to first fixation

As shown in Table 7.26, the *FRA-GI* had the best performance regarding the time to first fixation for *Actor A*. It obtained an average of 1.446 seconds, followed by the *DSA* with 1.615 seconds and the *FRA-FI* with 2.393 seconds. According to this, *Actor A* was located faster using the *FRA-GI*. Nevertheless, the difference between the evaluated approaches was found not to be statistically significant as illustrated in Table 7.27.

Table 7.25.: T-test for the fixations on target appearing over actor C

	T-test		
	FRA-FI - FRA-GI	**DSA - FRA-FI**	**DSA - FRA-GI**
Z	-3.134b	-0.935c	-3.124c
p	0.002	0.35	0.002

Table 7.26.: Time to first fixation for actor A

Time to First Fixation - Actor A			
ID	**FRA-FI**	**FRA-GI**	**DSA**
1163866620	4.402	2.061	0.774
1169451400	2.380	2.340	1.209
3341759460	5.521	3.615	2.923
4291681437	3.907	1.638	1.007
4295024649	3.118	1.515	0.001
4328697256	3.650	0.359	1.527
4407069149	0.000	1.374	5.139
4999902580	0.000	0.000	2.844
5032930026	0.000	2.310	1.803
5308616338	1.872	1.037	0.844
6385045328	5.697	0.651	1.753
6536263994	2.722	1.594	0.943
670690418	6.297	1.749	0.000
698441774	0.000	0.949	0.216
7026726368	0.000	0.000	1.953
7344799308	1.196	2.213	1.916
7507269062	1.445	1.626	1.320
7814128462	0.000	0.000	1.509
8969515821	0.566	3.879	0.542
9849432742	5.095	0.000	4.086
Average	**2.393**	**1.446**	**1.615**
Standard Deviation	**2.202**	**1.122**	**1.312**

Table 7.27.: T-test for the time to first fixation over actor A

	T-test		
	FRA-FI - FRA-GI	**DSA - FRA-FI**	**DSA - FRA-GI**
Z	-1.586b	-1.344b	0.000c
p	0.113	0.179	1

Table 7.28.: Time to first fixation for actor B

Time to First Fixation - Actor B			
ID	**FRA-FI**	**FRA-GI**	**DSA**
1163866620	0.054	0.000	4.404
1169451400	0.949	3.961	3.828
3341759460	2.860	3.353	3.044
4291681437	2.076	0.524	4.912
4295024649	0.937	0.501	1.319
4328697256	2.194	0.794	4.702
4407069149	3.709	0.421	6.770
4999902580	0.000	0.000	2.683
5032930026	1.365	1.416	3.214
5308616338	2.380	1.107	2.949
6385045328	8.423	3.133	2.327
6536263994	1.242	3.487	3.989
670690418	3.331	1.173	2.529
698441774	0.412	2.276	2.857
7026726368	2.429	0.698	5.690
7344799308	1.687	0.000	0.994
7507269062	6.932	2.500	1.115
7814128462	2.212	0.354	2.759
8969515821	2.084	1.924	0.794
9849432742	6.445	2.187	3.664
Average	**2.586**	**1.490**	**3.227**
Standard Deviation	**2.270**	**1.276**	**1.577**

Table 7.28 emphasizes that the *FRA-GI* had the best performance regarding the time to first fixation for *Actor B*. It obtained an average of 1.490 seconds, followed by the *FRA-FI* with 2.586 seconds and the *DSA* with 3.227 seconds. According to these results, *Actor B* was located faster using the *FRA-GI*. This was found to be statistically significant in comparison to the *DSA* ($[Z = -2.949c, p = 0.003]$). There was no evidence suggesting that the *DSA* was superior to the *FRA-FI* or vice versa. Likewise, the *FRA-FI* was found not to be superior to the *FRA-GI* as illustrated in Table 7.29.

As shown in Table 7.30, the *FRA-GI* had the best performance regarding the time to first fixation for *Actor C*. It obtained an average of 0.884 seconds, followed by the *DSA* with 1.710 seconds and the *FRA-FI* with 2.721 seconds. According to this, *Actor C* was located faster using the *FRA-GI*. This was found to be statistically significant in comparison to the *FRA-FI* ($[Z = -3.340b, p = 0.001]$). Similarly, the *DSA* ($[Z =$

Table 7.29.: T-test for the time to first fixation over actor B

	T-test		
	FRA-FI - FRA-GI	**DSA - FRA-FI**	**DSA - FRA-GI**
Z	-1.932b	-1.307c	-2.949c
p	0.053	0.191	0.003

Table 7.30.: Time to first fixation for actor C

Time to First Fixation - Actor C			
ID	**FRA-FI**	**FRA-GI**	**DSA**
1163866620	3.084	1.000	1.352
1169451400	1.338	0.432	0.425
3341759460	6.373	0.000	2.186
4291681437	1.561	0.248	0.929
4295024649	0.975	0.290	0.000
4328697256	2.116	1.913	1.429
4407069149	2.258	1.231	6.119
4999902580	0.000	0.000	0.000
5032930026	1.187	0.000	0.834
5308616338	1.621	1.029	1.838
6385045328	3.538	0.815	1.017
6536263994	1.257	0.385	4.231
670690418	4.999	1.092	0.196
698441774	2.378	0.000	1.394
7026726368	4.389	3.308	2.618
7344799308	1.264	0.390	1.829
7507269062	6.844	0.072	4.681
7814128462	5.339	1.069	1.734
8969515821	0.636	2.273	0.429
9849432742	3.270	2.142	0.953
Average	**2.721**	**0.884**	**1.710**
Standard Deviation	**1.958**	**0.916**	**1.620**

$-2.949b, p = 0.022$]) was superior to the *FRA-FI*. There was no evidence suggesting that the *FRA-GI* was superior to the *DSA* or vice versa, as illustrated in Table 7.31.

Discussion

According to the model of visual stability, the dynamic graph drawing produced by the *FRA-FI* had an absence of graph drawing offset (GDO). This is because the algorithm assigns a fixed position to the actors in the Euclidean Space. However, the *FRA-FI* allows multiple actors to occupy the same position at different points in time, which affects the visual stability of the dynamic graph drawing. The vertex set drawing stability (VDS), the edge set drawing stability (EDS) and the vertex set neighborhood change

Table 7.31.: T-test for the time to first fixation over actor C

	T-test		
	FRA-FI - FRA-GI	**DSA - FRA-FI**	**DSA - FRA-GI**
Z	-3.340b	-2.294b	-1.610c
p	0.001	0.022	0.107

(VDNC) considerably increased their values, indicating that the drawing maintained more elements on the screen although the network was actually changing. The students positively received the visual stability of the dynamic graph drawing produced by the *FRA-FI* and mentioned that the changes were not distracting or noticeable. Nevertheless, the students were confused to observe multiple actors in the same position. This was confirmed by the eye-tracking metrics. The number of fixations, the duration of fixations, the duration of the scan path as well as the fixations on target obtained the worst results with the *FRA-FI*. This means that the recognition of elements takes a considerable amount of time, while the visual search presents a very low accuracy. Therefore, the flickering reduction algorithm in combination with the flickering index is considered not to be suitable for tracking actors or patterns in a dynamic network.

The *FRA-GI* assigns a fixed position to the actors in the Euclidean Space. As a result, the model of visual stability reflects an absence of graph drawing offset (GDO). Nonetheless, the approach allows multiple actors to occupy the same position at different points in time. This affected the visual stability of the dynamic graph drawing. The vertex set drawing stability (VDS), the edge set drawing stability (EDS) and the vertex set neighborhood change (VDNC) increased their values drastically, indicating that more elements are maintained on the screen despite the network was actually changing. Once again, the visual stability of the dynamic graph drawing was perceived positively by the students and they mentioned that the changes were not distracting or noticeable. Despite this fact, the students were confused to observe several actors in the same position. According to the eye-tracking metrics, the *FRA-GI* obtained the best performance in terms of the recognition of elements; a finding which was endorsed by the duration of the fixations. Yet, the fixations on target had their worst performance with the *FRA-GI*. This implies that the students were faster identifying non-desired elements. Thus, the flickering reduction algorithm in combination with the gap index is considered not to be suitable for tracking actors or patterns in a dynamic network.

The *DSA* also assigns a fixed position to the actors in the Euclidean Space, which reduces the graph drawing offset (GDO) of the model to zero. In addition, the *DSA* introduces a series of restrictions to prevent multiple actors from appearing in the same position at different points in time. This feature impacted the visual stability of the dynamic graph drawing. The vertex set drawing stability (VDS), the edge set drawing stability (EDS) and the vertex set neighborhood change (VDNC) moderately increased their values. This indicates that only some elements are maintained on the screen during the temporal navigation. The students perceived the changes in the dynamic graph drawing. However, despite its visual discontinuities, the students mentioned that the drawing was less confusing. This was the case because only a few actors appeared in the same position. According to the eye-tracking metrics, the *DSA* obtained a balanced performance in comparison to the *FRA-FI* and the *FRA-GI*. Furthermore, the *DSA* had the most accurate visual search. This was confirmed by the fixations on target. Therefore, the degree stabilization algorithm is considered to support a satisfying user experience along with an efficient visual search when tracking actors or patterns in a dynamic network.

7.4. Summary

This chapter presented the case study to evaluate the flickering reduction algorithm and the degree stabilization algorithm described in Chapter 6. For this purpose, a group of 20 students was requested to track actors in three different scenarios. An eye-tracking device was used to record their eye movements, a questionnaire was provided to gather information about their experience and the model-based metrics quantified the visual stability of the dynamic graph drawings.

Scenario 1 utilized the flickering reduction algorithm in combination with the flickering index. *Scenario 2* employed the flickering reduction algorithm in combination with the gap index, while *Scenario 3* used the degree stabilization algorithm. The dynamic network illustrated by these drawings was extracted from the developer mailing list of the Asterisk framework described in Chapter 5. The research hypothesis for the case study implied that *dynamic graph drawings with a moderate level of visual stability support an efficient visual search and a satisfying user experience when tracking actors in a dynamic network.*

The data collected from the questionnaires provided a better insight about the user experience. They evaluated the five criteria presented in Chapter 5. This information was further analyzed with a Likert scale [77] along with a card sorting technique [3]. Additionally, the information recorded by the eye-tracking device was analyzed with metrics that have proven to quantify the efficiency of the visual search [26, 56, 57, 94]. Two settings were proposed for the analysis. The first setting applied the metrics to the overall drawing area and the second one where the three actors were located. These were the areas of interest (AOIs) for the case study.

The results obtained from the case study suggest that there is a trade-off between the efficiency of the visual search and the user experience, which depends on the visual stability of the dynamic graph drawing. On the one hand, placing multiple actors and relations in the same node and edge positions at different points in time increases the visual stability of the dynamic graph drawing. As a result, the user experience along with the time required to identify an element are improved. Nonetheless, the accuracy of the visual search decreases. On the other hand, introducing a restriction to prevent the positions to be overloaded provides the dynamic graph drawing a moderate level of visual stability. This makes the changes in the dynamic graph drawing to be more noticeable. Moreover, the user experience is slightly worsened along with the time required to recognize an element in the Euclidean Space. However, the accuracy of the visual search is considerably improved. In the following chapter, these findings are used as the basis to develop visually stable metaphors that allow the exploration of network attributes over time.

8. Exploring network attributes in visually stable representations

8.1. Introduction

As it has been reported in Chapters 5 and 7, the visual stability of a dynamic graph drawing affects the user experience and the efficiency of the visual search when tracking actors in a dynamic network. Based on this finding, it was possible to develop a visually stable representation to explore network attributes over time. It is named *Foresighted Heat Ring* since it combines the concepts of the foresighted graph layout [31], the heat map [17] and the radial layout [124].

The foresighted heat ring was used to track the productivity and collaborativeness of researchers in a dynamic author-publication network. In addition, the effectiveness of the technique was evaluated through a user study, which involved a group of 20 students. The result obtained suggest that visually stable representations provide a satisfying user experience when exploring network attributes over time. Furthermore, it is possible to utilize features such as highlighting or scaling [104, 106] to compensate the limited accuracy of the visual search.

The content of this chapter is based on the publication *"Foresighted Heat Ring: Tracking the Productivity and Collaborativeness of Researchers in Dynamic Networks"* [103].

8.2. The foresighted heat ring

The *Foresighted Heat Ring*, or *FHR*, is a visually stable representation designed to track the productivity and collaborativeness of researchers in a dynamic author-publication network. It distributes them around a circumference using a foresighted radial layout algorithm. The productivity and collaborativeness of these same researchers appear in the form of markers. The markers are placed on the heat map, which is located in the inner part of the FHR. Moreover, the tracking function of this visually stable representation is inspired by the theories of visual stability [83] and aims at improving the accuracy of the visual search. In the following, the details of the FHR are presented.

The foresighted radial layout

The foresighted radial layout is the algorithm in charge of placing on the canvas the researchers appearing the dynamic author-publication network. It computes a node partition as in the foresighted graph layout [31]. However, it differs from the original approach by ignoring the edge partition in order to prevent visual instability. For this purpose, seven mathematical definitions were developed. Two of them belong to

the original foresighted graph layout [31], while the other five were proposed to fit the requirements of the technique.

Definition 1 *A **graph** $g = (V, E)$ consists of a set of nodes V and a set of edges E, such that $E \subseteq V \times V$. In addition, $V(g)$ is used to denote the nodeset of graph g.*

Definition 2 *A **graph animation** G is a sequence $G = [g_1, \ldots, g_n]$ of graphs with $g_i = (V_i, E_i)$.*

Definition 3 *A **nodeset transition** NT is a sequence $NT = [V_1, \ldots, V_n]$ of nodesets based on a graph animation $G = [g1, \ldots, g_n]$, such that for all $i \in \{1, \ldots, n\}$: $V_i = V(g_i)$. Nodes can be added or removed with each transition.*

Definition 4 *Let $NT = [V_1, ...V_n]$ be a nodeset transition, then the **super nodeset** SN of NT is defined as $SN = SN(NT) = \bigcup_{i=1}^{n} V_i$.*

Definition 5 *Let $NT = [V_1, ...V_n]$ be a nodeset transition and $SN = SN(NT)$ its super nodeset. Then $T(v) = \{i | v \in V_i\}$ is the **lifetime** of the node $v \in SN$.*

Definition 6 *Let V be a nodeset and $V^* \subseteq P(V)$. A set V^* is an **partitioning** of V if and only if the elements in V^* are disjoint and $\bigcup_{v \in V^*} v = V$.*

Definition 7 *Let $NT = [V_1, ...V_n]$ be a nodeset transition and $SN = SN(NT)$ the super nodeset of NT. A partitioning $V^* = P_1, \ldots, P_k$ of SN is a **transition partitioning** of NT if and only if $v, v' \in P_r \Rightarrow T(v) \cap T(v') = \emptyset$.*

As illustrated in Figure 8.1, the foresighted radial layout starts by detecting all the researchers appearing in the dynamic author-publication network. This step is equivalent to build the *super nodeset* mentioned in the previous definitions. Afterward, the algorithm instantiates a matrix in order to store the *lifetime* of the researchers. The columns of the matrix represent the time periods of the dynamic author-publication network and the rows stand for the researchers. The foresighted radial layout places the value of 1 to indicate the appearance of a researcher or a 0 to denote his absence in the positions a_{ij} of the matrix. Figure 8.2 exemplifies this procedure.

Once the matrix has been initialized, the algorithm proceeds to compute the *transition partitioning* of the FHR. As illustrated in Figure 8.3, a new component is instantiated and the first researcher of the super nodeset is placed in this location. Since the algorithm is based on the foresighted graph layout, it is necessary to find candidate researchers that can be placed in the same component. The main condition is that a candidate researcher must have a disjoint lifetime with the researchers inserted in the component. If this condition is satisfied, the candidate researcher is placed into the component. Otherwise, the search process is continued. This process is repeated until all the researchers available in the super nodeset have been assigned to a unique component. As a result, the transition partitioning of the FHR is obtained. The last step of the algorithm consists of distributing the components of the transition partitioning in the outer part of the FHR. Figure 8.4 exemplifies the procedure and Figure 8.5 shows an overview of the algorithm.

```
1: function extractResearchers(networks)
2:     researchers ← new Researcher[];
3:     for i ← 0, networks.length do
4:         for j ← 0, networks[i].nodeset.length do
5:             r ← networks[i].nodeset[j];
6:             if !researchers.contains(r) then
7:                 researchers.insert(r);
8:             end if
9:         end for
10:    end for
11:    return researchers;
12: end function
```

Figure 8.1.: Function to extract the set of unique researchers

```
1: function buildLTM(networks, researchers)
2:     int[][]LTM ← new int[networks.length][researchers.length];
3:     for i ← 0, networks.length do
4:         for j ← 0, networks[i].nodeset.length do
5:             rIndex ← researchers.indexOf(networks[i].nodeset[j]);
6:             LTM[i][rIndex] = 1;
7:         end for
8:     end for
9:     return LTM;
10: end function
```

Figure 8.2.: Pseudo-code to build the matrix with the lifetimes of the researchers

```
1: function BUILDPARTITION(LTM, researchers)
2:     components[] ← new Component[];
3:     component ← new Component();
4:     components.insert(component);
5:     for i ← 0, researchers.length do
6:         currentResearcher ← researchers[i];
7:         for j ← 0, components.length do
8:             currentComponent ← components[j];
9:             if currentComponent.contains(currentResearcher) then
10:                currentComponent.insert(currentResearcher);
11:                break;
12:            else if
    disjointLifeTime(currentResearcher, currentComponent, LTM) then
13:                currentComponent.insert(currentResearcher);
14:                break;
15:            else
16:                newC ← newComponent();
17:                newC.insert(currentResearcher);
18:                components.insert(newC);
19:                break;
20:            end if
21:        end for
22:    end for
23:    return components;
24: end function
```

Figure 8.3.: Pseudo-code to build the node partition

```
1: function BUILDRADIALLAYOUT(components)
2:     for i ← 0, components.length do
3:         currentComponent ← components[i]
4:         angle ← 2 * PI * i/components.length
5:         x ← (canvas.width/2) + sin(angle) * (radius)
6:         y ← (canvas.height/2) + cos(angle) * (radius);
7:         currentComponent.setCoordinates(x, y)
8:     end for
9:     return components
10: end function
```

Figure 8.4.: Pseudo-code to build the radial layout

```
1: function FORESIGHTEDRADIALLAYOUT(networks)
2:     researchers ← extractResearchers(networks);
3:     LTM ← buildLTM(networks, researchers);
4:     partition ← buildPartition(LTM, researchers)
5:     partition ← buildRadialLayout(partition);
6:     return partition ;
7: end function
```

Figure 8.5.: Pseudo-code of the foresighted radial layout algorithm

The heat map

A heat map [17] is located in the inner part of the FHR. It displays a set of markers that represent the productivity and collaborativeness of the researchers appearing in the dynamic author-publication network. The size of a marker depends on the productivity. Researchers with higher productivity appear with larger markers, while researchers with lower productivity appear with smaller ones. The position of a marker on the heat map depends on the collaborativeness. Researchers with higher collaborativeness place their markers in the hot zone. It is located in the center of the FHR colored with a gradient from red to orange. Researchers with a moderate collaborativeness place their markers in the intermediate zone. It is located immediately after the hot zone colored with a gradient to the blue color. Finally, researchers with lower collaborativeness place their markers in the cold zone. It is located in the outer part of the FHR colored in blue.

Scientific productivity and collaborativeness

In social network analysis, an affiliation network is a particular type of graph in which a set of actors are exclusively connected to a set of events or activities. An author-publication network can be considered as an instance of such networks, in which the researchers/authors are the actors, the publications are the events and the links between them indicate a researcher/author has been involved in the creation of a publication. In the context of the FHR, the *degree centrality* [44] of the researchers/author has been utilized as an indicator for productivity [41, 75]. In contrast, as an indicator for collaborativeness, the degree centrality of the actors in the co-authorship network has been used. This network is derived from the author-publication network, applying a *graph folding operation* [117]. The FHR is not restricted to a specific metric for describing the productivity or collaborativeness of a researcher/author. For example, a metric such as *cooperativeness* [62] is calculated based on how active a researcher/author and his connections are over the years. This metric could be used as a viable replacement for the collaborativeness.

The productivity and collaborativeness of a researcher are calculated using an *analysis agent* of the web-based analytics workbench [54, 55] as described in Chapter 3. The *productivity and collaborativeness agent* receives as input a set of graphs encoded with the *SISOB Graph Format*. Each graph represents a snapshot of a dynamic author-publication network and for every snapshot the agent performs two operations. First, it calculates the *degree centrality* [44] of the researchers in the author-publication network. The resulting values are used as an indicator for the productivity. On the other hand, the agent executes a *graph folding operation* [117] on the author-publication network in order to obtain the co-authorship network. Then, it calculates the *degree centrality* [44] of the researchers co-authorship network and uses the resulting values as an indicator for the collaborativeness. The researchers are decorated with their productivity along with their collaborativeness. Once the complete dynamic author-publication network has been processed, the decorated network is returned to the *SQLSpaces* [119], where the next agent can access data.

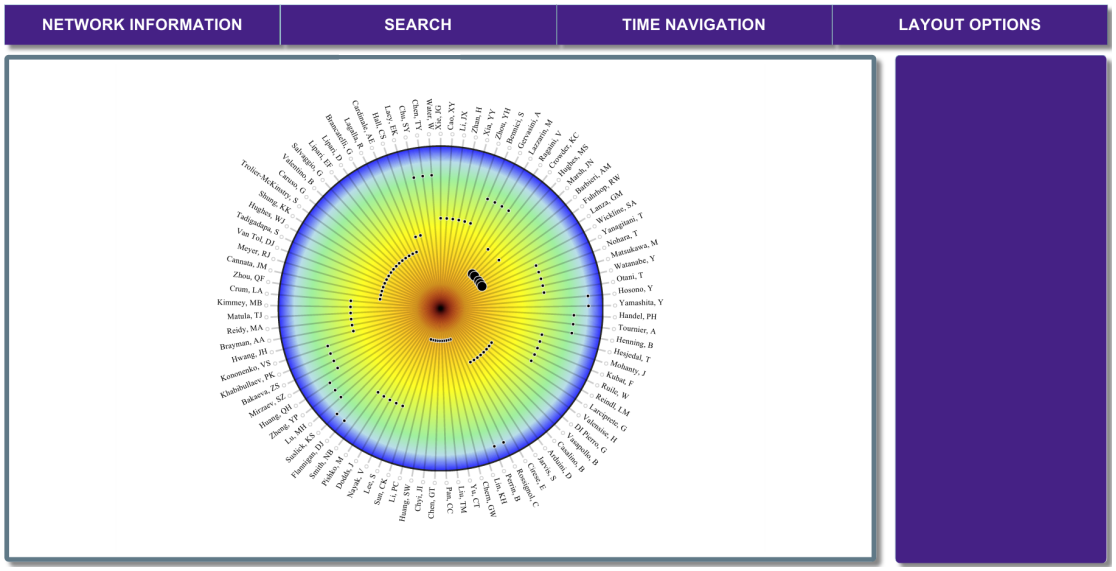

Figure 8.6.: Foresighted heat ring visualization

Features of the foresighted heat ring

As illustrated in Figure 8.6, the FHR presents three main elements on its user interface: the *menu bar*, the *toolbar* and the *drawing area*. The menu bar, located on the upper part of the screen, offers users the possibility to switch between different perspectives such as *network information*, *search*, *time navigation* and *layout options*. On the right-hand side, we find the toolbar. This element adjusts its content depending on the perspective selected by the user.

For example, the network information perspective displays the title of the dynamic author-publication network along with the period of time under exploration. The search perspective displays an input field, where the users can type the name of a specific researcher. The results obtained are listed below the input field and include the name of the researcher, their productivity along with their collaborativeness. At the same time, the FHR enlarges all these researchers on the drawing. Figure 8.7 exemplifies the search perspective.

The time navigation perspective displays a set of controls to explore the different periods of time that form the dynamic author-publication network. The *time frame selector* allows the user to introduce the period of time they want to visualize. As a result, the FHR is updated with the corresponding information. The *movie control* permits the user to observe the complete dynamic author-publication network as a movie clip without further interaction. The FHR introduces a time delay to perform a transition between two consecutive periods of time, which can be adjusted using *movie frame delay*. Figure 8.8 illustrates the time navigation perspective.

The layout options perspective displays a set of controls to adjust the elements appearing in the FHR. For instance, the *radius* option allows the user to increase or decrease the radius of the layout. The *label size* option permits the user to modify the size of the labels illustrating the names of the researchers. Finally, the *reset tracking* option restores the

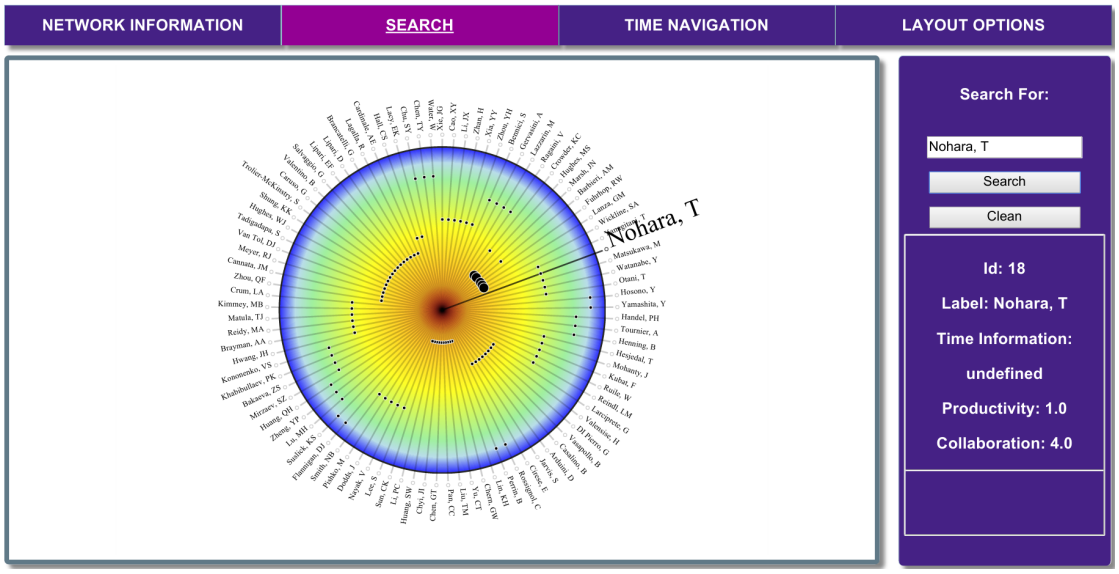

Figure 8.7.: Foresighted heat ring - search perspective

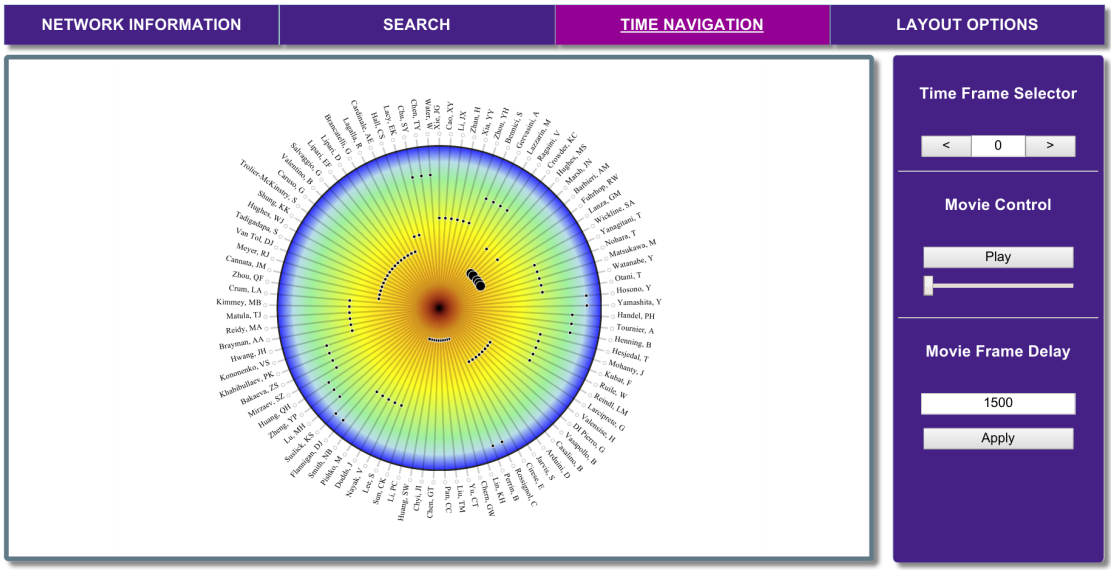

Figure 8.8.: Foresighted heat ring - time navigation perspective

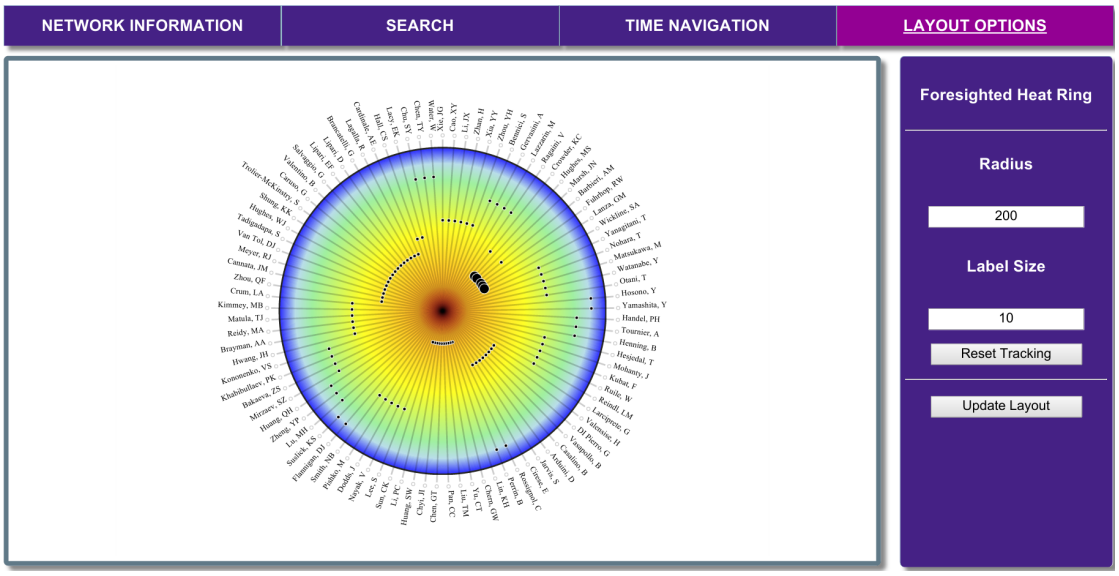

Figure 8.9.: Foresighted heat ring - layout options perspective

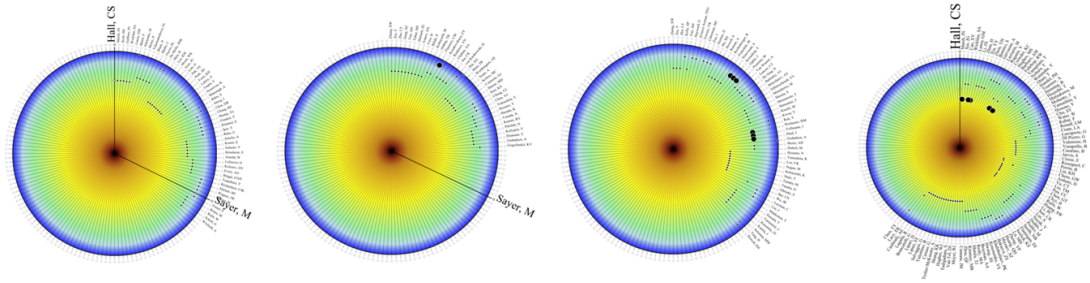

Figure 8.10.: Foresighted heat ring - tracking function

default configuration of the tracking function available in the FHR. Figure 8.9 exemplifies the layout options perspective.

The tracking function of the FHR is activated by selecting one of the labels in the drawing. The size of the label is considerably enlarged and the color of the marker changes to be differentiated from the others. As the user navigates through the dynamic network, the position of the label, its size, along with the color of the marker as always maintained. Furthermore, the use of a foresighted algorithm allows the FHR to minimize the movement of the other elements. There might be scenarios in which the selected researcher disappears from the dynamic network. In such cases, the FHR restores the label and the marker to their default configuration. However, if the selected researcher reappears, the FHR will increase the size of the label and change the color of the marker one more time. Figure 8.10 exemplifies the tracking function of the FHR.

Integration with the web-based analytics workbench

The FHR has been integrated into the web-based analytics workbench [54] using a hybrid visualization agent from the framework to support flexible visualization techniques described in Chapter 3. On the server, the visualization agent fetches the dynamic

author-publication network, which has already been decorated with the productivity and the collaborativeness of the researchers. The agent scans the dynamic network, looking for the maximum value of both indicators. This is necessary to calculate the size of the markers along with their position on the heat map. After the values have been found the dynamic author-publication network is decorated with two additional attributes: *maximum productivity* and *maximum collaborativeness*.

Once the maximum productivity and collaborativeness have been found, the agent proceeds with the complex operations of the foresighted radial layout. In a first instance, it computes the super node set along with the matrix structure containing the lifetime of the researchers in the dynamic author-publication network. Then, a new component is instantiated and the first researcher of the super nodeset is placed in this location. Since the foresighted radial layout is based on the foresighted graph layout [31], the agent starts searching for candidate researchers that can be placed in the same component. The only condition is that a candidate researcher must have a disjoint lifetime with the researchers inserted in the component. If this condition is satisfied, the candidate researcher is placed into the component. Otherwise, the search process is continued. This is repeated until all researchers available in the super nodeset have been assigned to a unique component. As a result, the transition partitioning of the FHR is obtained. The visualization agent creates a unique *ID* for every component in the transition partitioning and the researchers in the dynamic author-publication network are decorated with this information. Furthermore, the network is also decorated with the number of components that form the transition partitioning. Finally, the data, as well as the modules of the FHR, are deployed to the analysis result repository of the web-based analytics workbench.

On the client-side, the drawing engine accesses the maximum productivity and the maximum collaborativeness. The maximum productivity is used to create two different range of values. The first one is a range of productive values in the form [0, maximum productivity]. The second one is a range of scale values in the form [1, maximum marker size]. Both ranges are given to the D3.js[1] library, which returns a function to perform the mapping between the productivity and the size of the marker. The maximum collaborativeness is used to divide the radius of the FHR into an equal number of parts. The markers are placed on the section where the radius was divided depending on the value of the collaborativeness. After the size and the position of the markers have been calculated, the drawing engine proceeds to distribute the components of the transition partitioning on the canvas. It divides a circumference into an equal number of parts using the number of components in the transition partitioning. The components are placed on the section where the circumference was divided. At the same time, a label with the name of the researcher and the respective marker are displayed on the drawing.

8.3. Tracking productive and collaborative in a dynamic scientific network

The FHR was used to track the productivity and collaborativeness of researchers in a dynamic author-publication network from the field of nanotechnology. The researchers

[1] `http://d3js.org/`

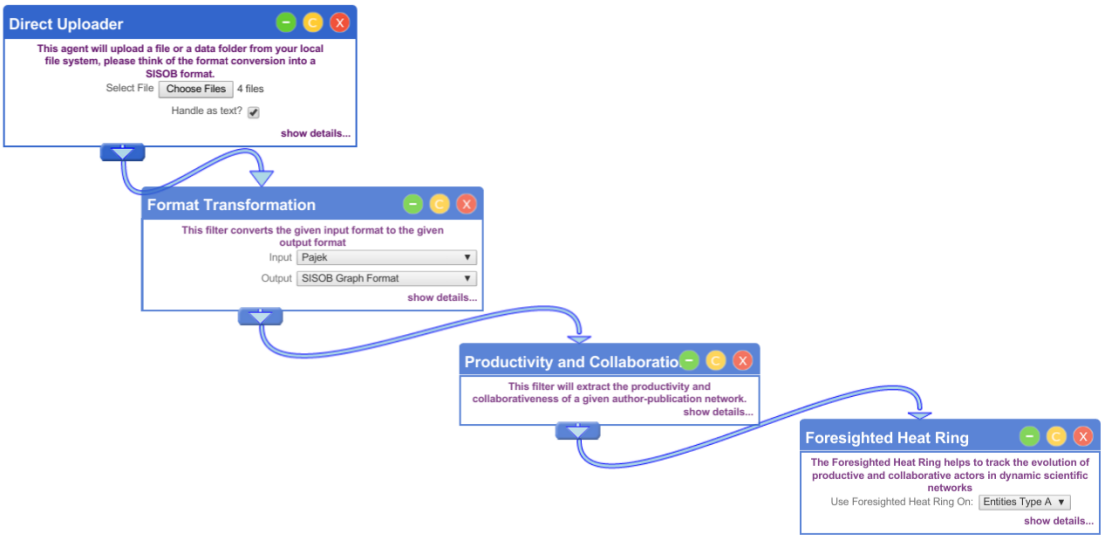

Figure 8.11.: Analysis workflow to visualize the productivity and collaborativeness of researchers using the foresighted heat ring

were members of the *Center of Nanointegration Duisburg-Essen*[2] (CENIDE). The dynamic network was extracted from *Web of Science*[3] in the form of a successive sequence of snapshots, which were captured using a time window [60] of two years over the period from 2000 to 2011.

A network analysis workflow was designed using the web-based analytics workbench [54, 55]. It aimed to extract the productivity and collaborativeness of the researchers, while the results were visualized with the FHR. As illustrated in Figure 8.11, four different filters were involved in the analysis:

- *Direct Uploader* - Filter responsible for uploading the dynamic author-publication network.
- *Format Transformation* - Filter responsible for transforming the dynamic author-publication network encoded in a Pajek.net format to the SISOB Graph Format.
- *Productivity and Collaborativeness* - Filter responsible for extracting the productivity and collaborativeness of the researchers appearing in the dynamic author-publication network.
- *Foresighted Heat Ring* - Filter responsible for executing the foresighted heat ring visualization.

As the dynamic network was explored, the movement of the markers in the FHR revealed an interesting phenomenon. As shown in Figure 8.12, the researcher *GS* appears for the first time in the snapshot from 2003-2005, with a productivity of 5 and a collaborativeness of 0. In the snapshot corresponding to the period between 2006-2008, the researcher increased his productivity and collaborativeness to 7 and 3, respectively. In the last snapshot of the sequence, the researcher has not only increased the productivity to 8 but

[2] http://www.uni-due.de/cenide/
[3] http://webofknowledge.com

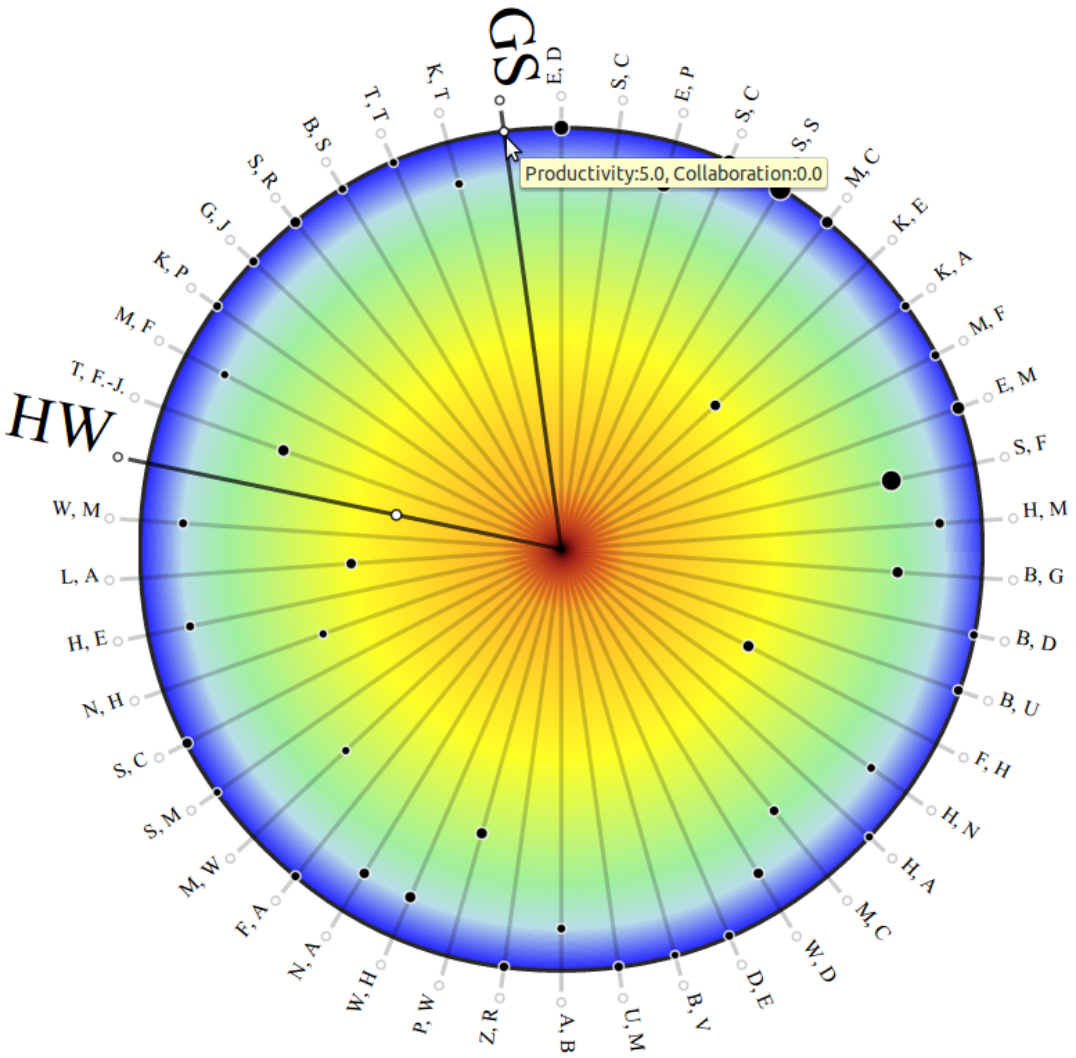

Figure 8.12.: First appearance of the researcher GS in the dynamic author-publication network

also appears near the center of the FHR with a collaborativeness of 4. This last change positioned the researcher as one of the most collaborative members of the CENIDE in that period of time.

In contrast to that, the researcher *HW* presented a different pattern. The researcher appears for the first time in the snapshot from 2000-2002 with a productivity of 4 and a collaborativeness of 0. In the snapshot corresponding to the period between 2003 to 2005, the researcher increased the productivity to 13 and the collaborativeness to 6. In the next snapshot, HW once again increased the productivity along with the collaborativeness. The researcher presented a productivity of 24, while the collaborativeness was 8. In the last snapshot of the sequence, HW increased the productivity to 40. Nonetheless, the collaborativeness decreased to 7. This finding is illustrated in Figure 8.13

The members of the CENIDE shared additional information about the discovery. GS was a young researcher who joined the CENIDE in the year 2003. Since then, the re-

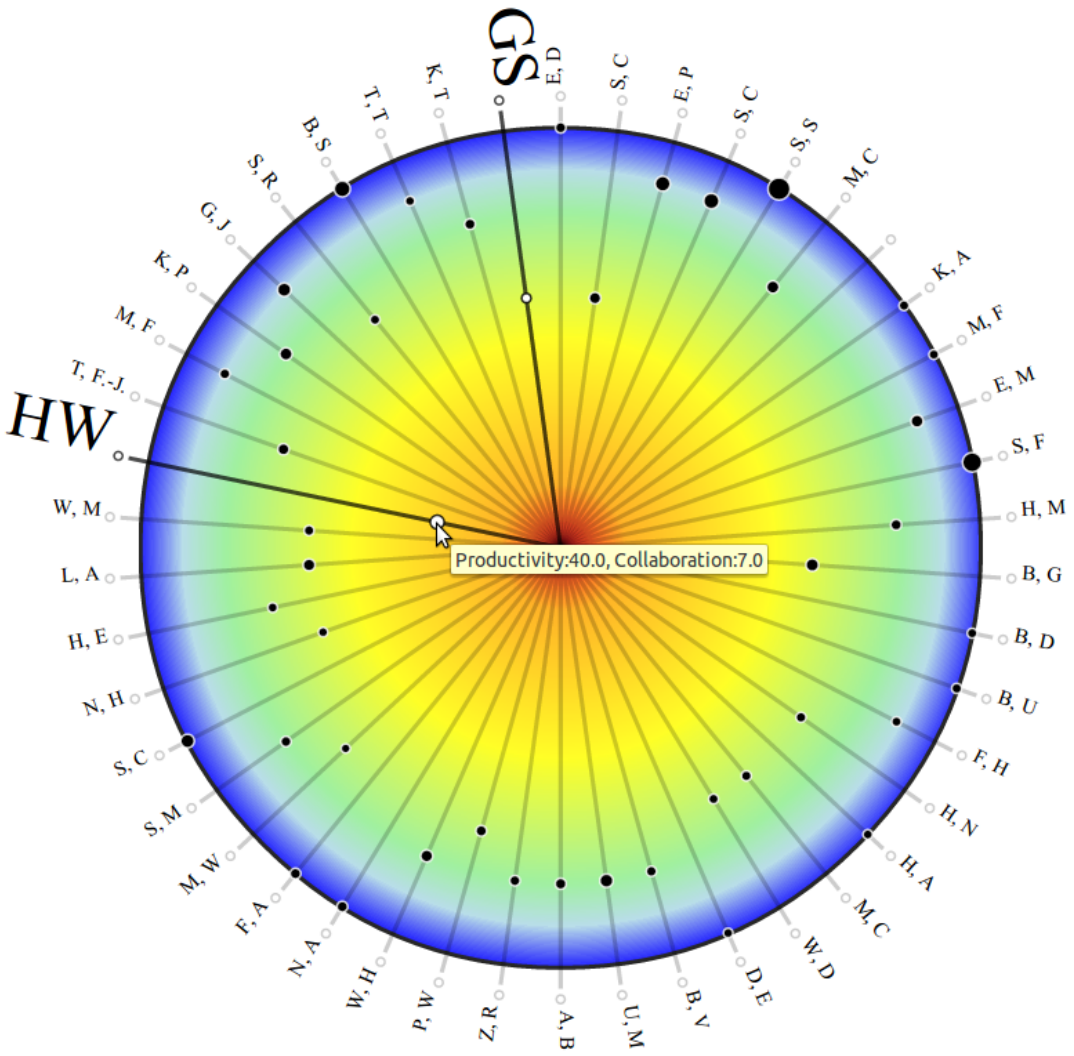

Figure 8.13.: Performance of the researcher HW in the last snapshot of the dynamic author-publication network

.

searchers has been very productive and collaborative, gaining a good position among the colleagues. The story behind HW is completely different. The researcher appears to be one of the experts in the field of nanotechnology because the productivity along with the collaborativeness changed quite drastically. However, HW only provides nanoparticles to other members of the CENIDE. Thus, the researcher always appears in the author list of several publications.

8.4. Evaluation of the technique

Study setup

A group of 20 students aged between 22 and 28 participated in the user study to evaluate the FHR. They were from the fields of mechatronics, electronics, physics, cognitive science and applied computer science participated. All students had a basic knowledge about social network analysis techniques but were not familiar with the theories of visual stability nor the mental map.

The students had to complete 15 tasks. These tasks were designed to evaluate the usability and functionality of the FHR. The first pair of tasks requested the students to locate certain components of the user interface, such as the menu bar, its options, and the canvas. Tasks 3 to 5 asked the students to interact with the labels surrounding the FHR. Task 6 requested the students to select one of the markers. Tasks 7 and 8 asked about the size of the markers, while tasks 9 and 10 about their movement inside the FHR. Lastly, tasks 11 to 15 requested the users to track two researchers in a dynamic author-publication network from the field of nanotechnology. These researchers were GS and HW.

At the end of the tasks, the students were requested to answer a questionnaire about their experience with the FHR. Each question was rated with a Likert scale of five points; where the value of 1 exemplified a strong disagreement, the value of 3 neither a disagreement nor an agreement, and the value of 5 a strong agreement. The criteria subject to evaluation were organized in two different groups. Positive criteria reflect better results with higher values on the Likert scale, while negative criteria reflect the opposite. The criteria subject to evaluation are listed below:

- C1 - The understanding of the visual representation.
- C2 - The selection of elements surrounding the radial layout.
- C3 - The additional information about the markers over the heat map.
- C4 - The understanding of the size of the nodes.
- C5 - The understanding of the heat scale.
- C6 - The color of the marker when tracking a researcher.
- C7 - The usefulness of the technique for tracking the productivity and collaborativeness of researchers.
- C8 - The distraction introduced by the movement of the elements surrounding the radial layout.

- C9 - The distraction introduced by the movement of the elements inside the radial layout.

There was no time limit for completing the tasks or the questionnaire. Still, the average duration per student for the study was 30 minutes. The tasks along with the questionnaire used in the user study are available in Appendix B.

Hypothesis

The research hypothesis for the user study implied that *visually stable representations support a satisfying user experience when tracking actors in a dynamic network.*

8.5. Analysis and results

The results obtained from the tasks indicated that the participants were able to visualize the researchers GS and HW without major problems. At the same time, the observations written by the participants revealed the same pattern described in Section 8.3. The researcher GS was identified in the network snapshot from 2003-2005 for the first time. Afterward, the researcher increased his productivity and collaborativeness between the years 2006 and 2008. By 2011, GS was one of the most collaborative members working at the CENIDE. The researcher HW was detected in the network snapshot from 2000 and 2002 for the first time. The participants of the study reported incremental changes in productivity and collaborativeness of the researcher, with a more drastic variation in the year 2011. Due to this observation, the participants thought that the researcher HW was one of the experts in the field of nanotechnology.

The results obtained from the questionnaires provided a better insight about the usefulness of the FHR. The average value of the user ratings per criterion was computed and the ratings lied withing a range of [1, 5] with a mean value of 3. Figure 8.14 illustrates the average user ratings from Criteria 1 to Criteria 7. These criteria belonged to the positive group, which means that a rating of 5 is perceived as the best while a rating of 1 is perceived as the worst. Figure 8.15, illustrates the average user ratings for Criteria 8 and 9. These criteria belonged to the negative group, which means that a rating of 5 is the worst while a rating of 1 is the best.

According to the results, the participants of the study were able to understand the concepts behind the FHR (C1 = 4.000). The name of the researchers was displayed in the outer part of the visualization, while the markers representing their productivity and collaborativeness were located over a heat map in the center of the technique. The participants noticed that the size of the markers reflected the productivity of a researcher (C4 = 4.450). Similarly, it was noticed that the location of the marker over the heat map reflected the collaborativeness of the researcher (C5 = 4.400). The tracking function was also received positively. The participants were able to select the researchers displayed in the outer part of the FHR without any problems (C2 = 4.600). Moreover, the change in the color of the marker was understood as a selection (C6 = 3.350) and that it was possible to obtain additional information with a tooltip by placing the mouse on the desired marker (C3 = 3.850). In addition, the movement of the elements in the

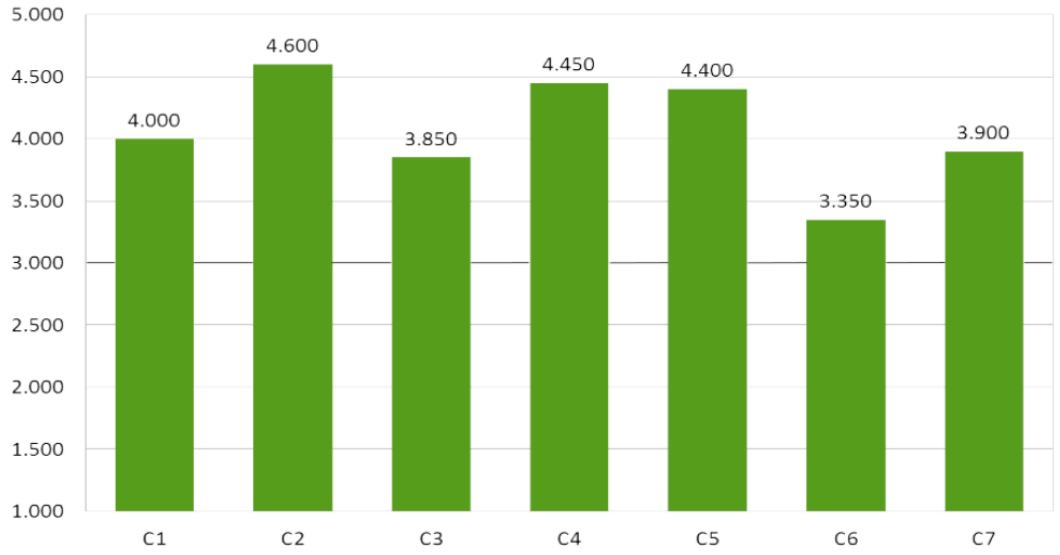

Figure 8.14.: Results of the positive criteria (C1, C7)

.

outer and inner part of the FHR was not distracting during the completion of the tasks (C8 = 1.700, C9 = 1.750). As a result, the participants agreed that the FHR is a useful technique for tracking the productivity and collaborativeness of researchers in a dynamic author-publication network (C7 = 3.900).

8.6. Discussion

As it has been mentioned in Chapters 5 and 7, the visual search achieves its highest levels of accuracy when each actor in the dynamic network is mapped to a unique position in the Euclidean Space. Nonetheless, the user experience is negatively affected due to the constant addition and removal of elements from the drawing. As an alternative, it is possible to assign multiple actors to the same position. This forces the drawing to achieve visual stability. As a result, the user experience considerably improves at the cost of losing accuracy on the visual search.

These findings were taken into account during the development of the FHR. It uses a variation of the original foresighted graph layout [31] because the algorithms introduced in Chapter 6 might not have achieved the desired objective, i.e., prove that visually stable representations are useful to explore network attributes over time. The foresighted radial layout guarantees that the drawing will be perceived as visually stable. At the same time, the heat map displays the desired network attributes, while the tracking function highlights them at any time.

The FHR was used to track the productivity and collaborativeness of two researchers in a dynamic author-publication network. Furthermore, its effectiveness was evaluated through a user study, assuming that visually stable representations allow the exploration of network attributes over time. The results obtained suggest that the users were able to

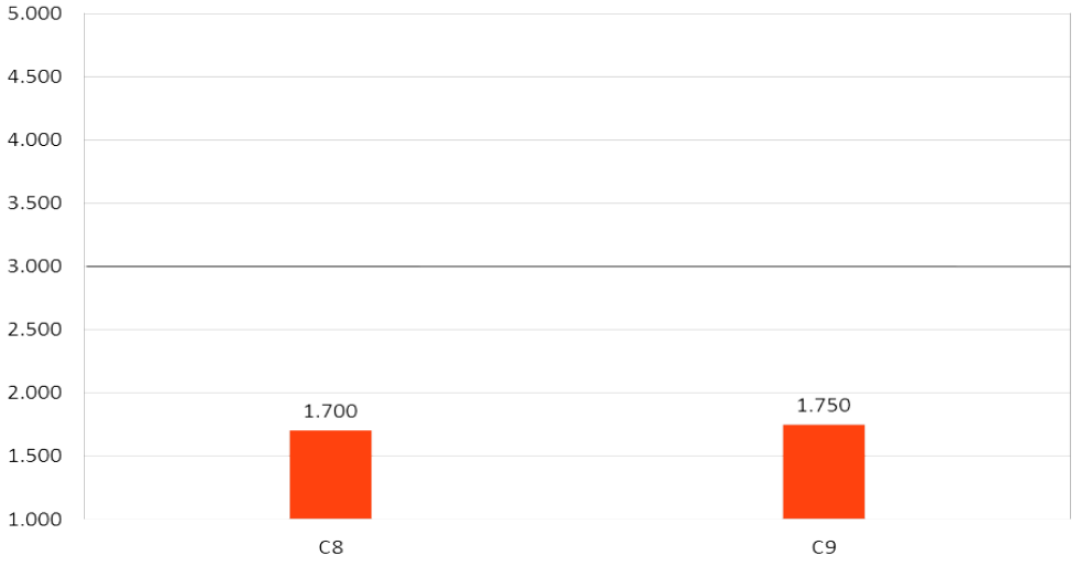

Figure 8.15.: Results of the negative criteria (C8, C9)

.

track the two researchers without major problems. The concept of the FHR, the notion of the markers along with the visual stability of the drawing was received positively. The movement of the markers located in the outer and inner parts of the FHR was not distracting at all. In addition, the tracking function allowed the participants of the study to easily locate the requested researchers. This indicates that visually stable representations can be enhanced with features such as the highlighting or scaling of elements [104, 106] to compensate the limited of accuracy on the visual search. At the same time, they provide a satisfying user experience as the attributes of a dynamic network are explored.

8.7. Summary

This chapter presented the foresighted heat ring, a visually stable representation for tracking the productivity and collaborativeness of researchers in an author-publication network. It receives its name from combining the concepts of the foresighted graph layout [31] with the heat map heat map [17] and the radial layout [124]. In addition, the foresighted heat ring incorporates a tracking function inspired by the theories of visual stability [83]. This last feature aims at improving the accuracy of the visual search.

The foresighted radial layout is the algorithm responsible for placing on the canvas the researchers appearing in the dynamic network. It computes a node partition as in the foresighted graph layout but omits the edge partition to prevent visual instability in the drawing. A researcher is arbitrarily selected and placed into a new component. Afterward, a set of researchers who can share a position in the same component are located. The only restriction is that these researchers must have a disjoint lifetime with those that are already in the component. Such a procedure is repeated until every

researcher has been assigned to a component. Finally, the resulting components of the partition are distributed around a circumference.

The heat map is located in the inner part of the FHR. It displays a set of markers, which represent the productivity and collaborativeness of the researchers. On the one hand, the size of a marker depends on the productivity. Researchers with higher productivity present larger markers, while researchers with lower productivity present the opposite. On the other hand, the position of a marker on the heat map depends on the collaborativeness. Researchers with higher collaborativeness place their markers on the hot zone, those with a moderate level in the intermediate zone and those with lower collaborativeness in the cold zone.

The FHR was used to track the productivity and collaborativeness of researchers, working at the *Center of Nanointegration Duisburg-Essen*[4] in a dynamic author-publication network from the field of nanotechnology. It was extracted from *Web of Science*[5], capturing a successive sequence of snapshots from the years 2000 to 2011 with a time window of two years. The web-based analytics workbench was used to calculate the productivity and collaborativeness, while the FHR illustrated the results. It was integrated into this analysis platform with a hybrid visualization agent of the framework to support flexible visualization techniques described in Chapter 3.

When the dynamic author-publication network was explored, the movement of the markers in the FHR exposed a pattern. The researcher *GS* appeared in the snapshot from 2003-2005 for the first time with a productivity of 5 and a collaborativeness of 0. In the snapshot from 2006-2008, the researcher increased his productivity and collaborativeness to 7 and 3, respectively. In the last snapshot of the sequence, the researcher did not only increase the productivity to 8 but also appeared near the center of the FHR with a collaborativeness of 4.

In contrast, the researcher *HW* presented a different pattern. He appeared in the snapshot from 2000-2002 for the first time with a productivity of 4 and a collaborativeness of 0. In the snapshot from 2003-2005, the researcher increased his productivity to 13 and the collaborativeness to 6. In the next snapshot, *HW* increased the productivity along with the collaborativeness once again. Both indicators reached the values of 24 and 8, respectively. In the last snapshot of the sequence, *HW* had a productivity of 40. Nonetheless, the collaborativeness decreased to 7.

According to additional information provided by the CENIDE, the researcher *GS* was a young person who joined the group in the year 2003. Since then, he has been very productive and collaborative, gaining a prestigious position among his colleagues. The story of *HW* is completely different. The researcher seemed to be one of the experts in the field of nanotechnology due to the changes on the productivity and collaborativeness. However,he only provided nanoparticles to the other members of the CENIDE. Thus, *HW* appears in the author list of several publications.

The previous findings were used in a study to evaluate the effectiveness of the FHR. It involved questionnaires to gather information about the user experience when tracking the researchers *GS* and *HW*. The research hypothesis for the study implied that *visually stable representations support a satisfying user experience when tracking actors in a*

[4] `http://www.uni-due.de/cenide/`
[5] `http://webofknowledge.com`

dynamic network. According to the results, the participants of the study were able to understand the concept of the FHR. The names of the researchers were found around the circumference. The markers on the heat map represented their productivity and collaborativeness. In addition, the size of a marker was interpreted as the productivity, while its location on the heat map as the collaborativeness. The tracking function of the FHR was received positively among the participants. They were able to track the researchers *GS* and *HW* without any problems. At the same time, the movement of the elements in the outer and inner parts of the FHR was not distracting during exploration of the dynamic network. This proves that visually stable representations are useful to explore network attributes over time, despite their limited accuracy on the visual search.

9. Conclusions

This thesis presented the development of visualization tools for dynamic network and the evaluation of visual stability characteristics. A framework to support flexible visualization techniques was developed for this purpose. It addresses the technical limitations of the existing tools to visualize dynamic networks, such as the performance of the drawing engine and their feature extensibility. The framework to support flexible visualization techniques served as the platform to study how the visual stability of dynamic graph drawings affect the efficiency of the visual search and the user experience when tracking actors or other patterns over time. The following chapter presents the summary of the thesis. Furthermore, it discusses the main achievements and possible future directions to study visual stability in dynamic graph drawings.

9.1. Summary of the thesis

In recent years, there has been a growing interest among researchers to study dynamic networks. These graph models have the characteristic of adding or removing actors and their relations at different points in time. Dynamic networks are based on three main concepts. First, the timeline. It refers to the overall period of time that contains the recorded information about the network [14]. Second, the time window. It is an interval whose start and end points specify the period of time in which the network will be under observation [60]. Lastly, the representation of time. This last one can be continuous or discrete [14].

The continuous representation of time [14] considers the actors and their relations as events that appear all over the network timeline. Such events have an exact starting and ending time, whose duration simulate a "stream" of information. The time window [60] is used to capture the elements in the stream. Thus, all those actors and relations whose duration fall within the time window form the dynamic network.

In contrast, the discrete representation of time [14] considers that the actors and their relations exist only at a certain point in time rather than having a time duration. The time window [60] is used to capture a successive sequence of "states" or "snapshots" from the network timeline. As a result, all those actors and relations captured in the snapshots form the dynamic network. The approach described beforehand is frequently utilized in the analysis and visualization of dynamic networks.

When int comes to the visualization of dynamic networks, dynamic graph drawings are the metaphor of choice among researchers. These drawings inherit the characteristics of the network under study since they use the same representation of time. The render window [88] acts as a direct reflection of the time window [60]. It illustrates the actors and relations that are available at a certain point in time. Hence, dynamic graph drawings created with the continuous representation of time can illustrate partially or completely

the stream of events; whereas those created with the discrete representation of time illustrate the different states of the dynamic network.

Nowadays dynamic graph drawings are created with the discrete representation of time because this approach is the most utilized in the analysis of dynamic networks. For each "snapshot" a graph drawing is computed with the algorithm of choice and the resulting sequence is presented to the user in a predefined order. Despite the simplicity of the method, the dynamic graph drawings that have been created with the discrete representation of time present some issues.

The graph drawing algorithm computes a distinct layout for each "snapshot" of the dynamic network. As a consequence, it is very likely for actors, relations or patterns to change their position on the canvas as the dynamic network is explored. Furthermore, dynamic graph drawings are constantly adding or removing nodes and edges at different points in time. The combination of these factors makes it very difficult to track something over time and can disturb the user to the point of breaking the mental map.

The term mental map refers to *"the structural cognitive information a user creates internally by observing the layout of a graph"* [32]. It was introduced in the context of interactive graph drawings, where researchers like Eades [37], Böhringer [18] and Messinger [85] evaluated it for the first time. According to them, minimizing the changes in a graph drawing helps to preserve the mental map. However, how to appropriately minimize such changes is still being discussed among the members of the scientific community.

Modern software applications to visualize dynamic networks combine graph drawing algorithms with sophisticated layout adjustment techniques. These technique minimize the changes in a dynamic graph drawing, either by *maintaining a constant shape* [7], *restricting the movement of the nodes and edges* [40], or *reducing their addition and removal* from the canvas [31]. These characteristics are mentioned in the theories of visual stability which are related to the origin of the mental map.

According to them [25, 82, 83], each saccade attempts to direct the eye-gaze towards objects that capture our attention. These objects are scanned in detail, while a mental representation of their respective locations is generated at the same time. In other words, the mental map. As new information becomes available in the visual field, the next saccades perform a search in the approximate region where the scanned objects were located. Thus, the sense of a stable image occurs it the objects are found.

The principles established in the theories of visual stability raise a question regarding the visualization of dynamic networks:

Does the visual stability of a dynamic graph drawing improves the efficiency of the visual search and the user experience when tracking actors or other patterns in a dynamic social network?.

In order to answer this question, it was necessary to address some technical and algorithmic limitations.

Technical limitations

The existing software applications to visualize dynamic networks are executed on a local machine or on a web server. On the one hand, those executed on a local machine incorporate distinct dynamic graph drawing algorithms, [52], layout adjustment techniques [7], and features to enhance the respective drawings [10]. Moreover, they support a variety of formats to encode network information. On the other hand, those executed on a web server are usually mounted on top of large databases and extract different types of networks. In addition, the extracted networks are illustrated with more complex metaphors rather than just graph drawings [47].

Both types of software applications have useful characteristics to visualize dynamic social networks. Nonetheless, they also have their own technical limitations. For example, a first technical limitation appears when visualizing large networks. The drawing engine is the component responsible for computing the visual representation of the dynamic network [10] and its performance depends on the resources of the computer. In the case of those software applications that are executed on a local machine, the performance of the drawing engine decreases as the data volume becomes larger. Certainly, those software applications that run on a web server have more resources. Unfortunately, the drawing is always computed on the client-side and only in rare occasions it has access to the resources of the server. As a consequence, the rendering of the drawing becomes slower as the size of the data increases.

A second technical limitation is related to the extensibility of these software applications. Those executed on a local machine present well-defined interfaces and allow the integration of new layout adjustment techniques or more sophisticated visual metaphors [10]. However, the performance of the drawing engine might not be the adequate to study the implications of visual stability. The software applications executed on a web server have more resources at their disposal but do not have well-defined interfaces to integrate new approaches nor instructions to adjust the performance of the drawing engine [47, 62].

Algorithmic limitations

The algorithmic limitations concert the existing metrics to quantify the changes in a dynamic graph drawing. Approaches like the Euclidean Distance and the Manhattan Distance assume that each actor in the dynamic network is mapped to a unique position on the canvas. However, some recent developments suggest that not all dynamic graph drawing algorithms use such a mapping.

Diehl [31] developed a dynamic graph drawing algorithm known as the *Foresighted Graph Layout*. It is able to map multiple actors to the same node position and multiple relations to the same edge position on the canvas at different points in time. Furthermore, the Foresighted Graph Layout can be utilized in conjunction with distinct graph drawing algorithms or layout adjustment techniques. As a result, it is impossible for the existing metrics to quantify the changes in such a drawing.

Solutions

A framework to support flexible visualization techniques

The framework to support flexible visualization techniques was developed during the run of the European Union project SISOB[1]. It addresses the technical limitations of the existing software applications to visualize dynamic networks. For example, the framework allows graph drawing algorithms, layout adjustment techniques and different types of visual metaphors to be integrated into a single analysis platform, i.e., the web-based analytics workbench [54]. In addition, the drawing engine of the framework can adjust its behavior using the multi-agent system of the workbench. Client-side visualization agents delegate the processing of the drawing to the engine on the web browser. Server-side visualization agents process the entire drawing on the server and send the resulting information to the engine, which completes the visual representation on the client. Hybrid visualization agents combine the previous characteristics. They perform the heavy computations of the drawing on the server, while the drawing engine completes the visual representation on the client performing more simple operations.

The web-based analytics workbench [54] decorates the nodes and edges of the network under study with numeric or non-numeric attributes during the execution of an analysis workflow. The framework uses this information to modify the graph drawing based on the needs of the user. For instance, the filtering function can remove unnecessary elements from the canvas. The styling function can adjust the visual appearance of labels, nodes, and edges; while the search function can highlight the desired element or entire neighborhoods. The workbench uses the discrete representation of time to create dynamic networks. The framework follows the same approach to produce dynamic graph drawings and offers a time navigation function to explore the different periods of time. Some basic layout adjustment techniques have been integrated into the framework. However, more sophisticated approaches can be incorporated through the visualization agents. The framework to support flexible visualization techniques served as a platform to study the effects of visual stability in dynamic graph drawings.

A mathematical model of visual stability for dynamic graph drawings

The mathematical model of visual stability addresses the algorithmic limitations of the metrics to quantify the changes in a dynamic graph drawing. It considers that every actor in the dynamic network is mapped to a vertex logical position, which is mapped afterward to the Euclidean Space. The same principle applies to the relations. Every relation in the dynamic network is mapped to an edge logical position and then to the Euclidean Space. This intermediate step allows the model to accurately quantify the changes in a dynamic graph drawing, even if it is created with the Foresighted Graph Layout [31].

Nine metrics form the mathematical model of visual stability. Each one of them covers an aspect of the dynamic graph drawing:

[1] http://sisob.lcc.uma.es/

- *Vertex set drawing active positions* - Calculates the percentage of vertex logical positions that have been active in the dynamic network.
- *Edge set drawing active positions* - Calculates the percentage of edge logical positions that have been active in the dynamic network.
- *Graph drawing offset* - Calculates the changes on the coordinates of those vertex logical positions that have been mapped to the Euclidean Space.
- *Vertex set stability* - Calculates the percentage of vertices that remain constant after a transition between two consecutive periods of time.
- *Vertex set drawing stability* -Calculates the percentage of vertex logical positions that remain constant after a transition between two consecutive periods of time.
- *Edge set stability* - Calculates the percentage of edges that remain constant after a transition between two consecutive periods of time.
- *Edge set drawing stability* - Calculates the percentage of edge logical positions that remain constant after a transition between two consecutive periods of time.
- *Vertex set degree change* - Calculates the variations to the degree of those vertices that remain constant after a transition between two consecutive periods of time.
- *Vertex set drawing neighborhood change* - Calculates the variations on the number of connections for those vertex logical positions that remain constant after a transition between two consecutive periods of time.

A case study was conducted to evaluate the mathematical model of visual stability. It requested the participants to track three actors in three different dynamic graph drawings. These were characterized by a *constant shape*, the use of *fixed positions* and *minimal structural changes* during the temporal navigation. In addition, an eye eye-tracking device (SMI - RED500)[2] recorded the eye movements of the participants; a questionnaire gathered information about their experience, while the model-based metrics quantified the visual stability of the respective drawings.

The research hypothesis for the case study implied that: *visually stable dynamic graph drawings improve the efficiency of the visual search and the user experience when tracking actors in a dynamic network.*

According to the results, the visual stability of a dynamic graph drawing affects the efficiency of the visual search and the user experience when tracking actors in a dynamic network. Those dynamic graph drawings that change the position of the nodes to achieve a constant shape have an extensive visual search, which makes it difficult to track actors in a dynamic network. As an alternative, dynamic graph drawings can assign to each actor in the dynamic network a fixed position on the canvas. This improves the accuracy of the visual search. However, the user experience is negatively affected due to the constant addition and removal of elements from the dynamic graph drawing. A solution to this problem is to allow multiple actors to occupy the same position at different points in time. This reduces the constant addition and removal of elements from the dynamic graph drawing, which improves the user experience. Nonetheless, such a characteristic reduces the accuracy of the visual search.

[2]http://www.smivision.com

Algorithms to support visual stability in dynamic graph drawings

Two dynamic graph drawing algorithms were developed as a result of the previous study. Both are inspired by the Foresighted Graph Layout [31] and aim to support an efficient visual search along with a satisfying user experience when tracking actors in a dynamic network.

The nodes in a dynamic graph drawing can be considered as "bulbs", which alternate between the presence and absence of light. Such a characteristic is due to the lifetime of the actors in the dynamic network. Actors with longer lifetimes illuminate the bulbs for longer periods of time, while actors with shorter lifetimes do the opposite. However, the lifetime of the actors in a dynamic network tends to be discontinuous. As a consequence, the bulbs produce a flickering that affects the visual stability of the dynamic graph drawing.

The *Flickering Reduction Algorithm* was developed to minimize such an anomaly from the dynamic graph drawing. The algorithm assumes that the flickering is produced by those actors with discontinuous lifetimes. Therefore, it introduces two different indicators to detect them. On the one hand, the *Flickering Index* stands for the number of times an actor enters and exits the dynamic network. On the other hand, the *Gap Index* stands for the number of inactive time periods an actor presents in the dynamic network. The actors with the highest values on the selected index are used as the basis to compute a node partition like in the Foresighted Graph Layout [31], whereas the edge partition is computed as in the original algorithm. The *Flickering Reduction Algorithm* was integrated into the web-based analytics workbench [54] using a server-side visualization agent from the framework to support flexible visualization techniques.

The *Degree Stabilization Algorithm* was developed following a social network analysis perspective. It minimizes the changes on the connections between the nodes in the dynamic graph drawing. An actor is selected arbitrarily and is used as the basis to compute a node partition like in the Foresighted Graph Layout [31]. However, certain restrictions must be satisfied in order to create the component. All actors must have disjoint lifetimes. Furthermore, they must contribute to reduce the average degree gradient. This stabilizes the connections of the nodes that appear in the dynamic graph drawing and prevents the component to be overloaded with too many actors. In regards to the edge partition, it is computed as described in the original Foresighted Graph Layout [31]. The *Degree Stabilization Algorithm* was integrated into the web-based analytics workbench [54] using a server-side visualization agent from the framework to support flexible visualization techniques.

In order to determine which of the developed algorithms support an efficient visual search and a satisfying user experience when tracking actors in a dynamic network, it was necessary to conduct a case study. This combined the use of an eye-tracking device to record the eye movements of the participants; a questionnaire to gather information about their experience and the model-based metrics to quantify the visual stability of three dynamic graph drawings.

The first dynamic graph drawing was created with the *Flickering Reduction Algorithm* in combination with the *Flickering Index*. The second dynamic graph drawing was created with the same approach but in combination with the *Gap Index*. Lastly, the third dynamic graph drawing was created with the *Degree Stabilization Algorithm*. The

participants of the case study were requested to track three actors using the previous dynamic graph drawings.

The research hypothesis for the case study implied that *dynamic graph drawings with a moderate level of visual stability support an efficient visual search and a satisfying user experience when tracking actors in a dynamic network.*

According to the results, there is a trade-off between the user experience and the efficiency of the visual search which depends on the visual stability of a dynamic graph drawing. Among the evaluated approaches, the *Flickering Reduction Algorithm* presented the highest levels of visual stability. Such a characteristic minimizes the addition or removal of nodes and edges during the exploration of the dynamic network. As a result, the user experience is improved during the tracking tasks. However, the accuracy of the visual search is worsened. This is because placing a considerable number of actors in the same position at different points in time confuses the user. In contrast, *Degree Stabilization Algorithm* presented a moderate level of visual stability. Such a characteristic makes the changes in a dynamic graph drawing more noticeable but not enough to distract the user during the tracking tasks. Furthermore, the accuracy of the visual is improved. This is because the degree stabilization algorithm restricts the number of actors that can be placed into a node position. Thus, the user is able to locate with ease the desired element.

Visually stable representations to explore network attributes over time

The *Foresighted Heat Ring* or *FHR* is a visually stable representation which was developed as a result of the previous case studies. It is designed to track the productivity and collaborativeness of researchers in a dynamic author-publication network. The FHR presents three main characteristics. First, a foresighted radial layout. It distributes the researchers around a circumference. Second, a heat map [17, 79, 121]. It displays the productivity and collaborativeness of the researchers. Lastly, a tracking function whose features compensate the limited accuracy of the visual search.

The *Foresighted Radial Layout* is inspired by the Foresighted Graph Layout [31]. It computes a node partition as described in the original approach but omits the edge partition to prevent visual instability in the drawing. The foresighted radial layout proceeds as follows. An actor is arbitrarily selected from the dynamic network and it is immediately placed into a new component. Then, a set of actors who can share a position in the same component is searched. The only restriction is that these actors must have a disjoint lifetime with those in the component. Such a procedure is repeated until all the actors have been processed. Lastly, the resulting components are distributed around a circumference.

The *Heat Map* is located in the inner part of the FHR. It displays a set of markers which represent the productivity and collaborativeness of the researchers. The size of the marker stands for the productivity. Markers of larger size are assigned to researchers with higher productivity, while smaller markers are assigned to those with lower values. The position of the marker on the heat map stands for the collaborativeness. The markers of those researchers with higher collaborativeness are placed in a hot zone. It is located in the center of the visualization and colored with a gradient from red to orange.

The markers of those researchers with moderate collaborativeness are placed in the intermediate zone. This one is located after the hot zone and it is colored with a gradient from orange to blue. Lastly, the markers of those researchers with low collaborativeness are placed in the cold zone. It is located in the outer part of the visualization and colored in blue.

The productivity and collaborativeness of a researcher can be calculated using social network analysis techniques. For instance, the productivity is equal to the degree centrality of the researcher in the author-publication network. The collaborativeness is equal to the degree centrality of the researcher in the co-authorship network.The *FHR* is not restricted to a specific set of metrics to describe the productivity or collaborativeness of researchers. Approaches like cooperativeness [62] can be used as a replacement in case it is necessary.

The effectiveness of the *FHR* was evaluated through a study. A group of students was requested to track two members of the Center of Nanointegration Duisburg-Essen[3]. The dynamic author-publication network was from the field of nanotechnology. Its timeline was from the years 2000 to 2011 and four snapshots were captured. The web-based analytics workbench [54] was used to calculate the productivity and collaborativeness of the researchers. The *FHR* was used to illustrated the results. It was integrated into the workbench with a hybrid visualization agent of the framework to support flexible visualization techniques.

The research hypothesis for the study implied that *visually stable representations support a satisfying user experience when tracking actors in a dynamic network.*

The results obtained from the study suggest that the participants were able to track the requested researchers without problems using the *FHR*. They noticed that the size of a marker reflected the productivity of a researcher, while the location of the marker on the heat map reflected the collaborativeness. In addition, increasing the scale of the labels and the change on the color of the markers during the tracking tasks was received positively. Furthermore, the movement of the markers inside the foresighted heat ring was not distracting. These findings indicate that visually stable representations provide a satisfying user experience when tracking network related attributes over time.

9.2. Conclusions and future work

As it has been presented in this thesis, visual stability is a property that must be considered when tracking actors or other patterns in a dynamic network. Those dynamic graph drawings changing the position of the actors in the Euclidean Space to maintain a constant shape, do not provide a good user experience nor an efficient visual search. Users require more effort to locate the desired actors. This was confirmed through the information recorded by the eye-tracking device. In such drawings, the length of the scan paths was considerably larger, while the number of fixations on target was extremely low. Therefore, it is not recommended to change the position of the actors when tracking tasks are involved.

[3] http://www.uni-due.de/cenide/

In order to improve the performance of both parameters, it is possible to map to a fixed position in the Euclidean Space each actor appearing in the dynamic network. As a result, the users require less effort to locate the desired actors because they always appear in the same position. This was affirmed using the information recorded by the eye-tracking device. The scan paths were found to be considerably shorter, the number of fixations on target drastically increased, while the spatial density was very low. However, mapping each actor in the dynamic network to a fixed position in the Euclidean Space has a disadvantage. Nodes and edges are constantly added or removed at different points in time which makes the dynamic graph drawing to be perceived as visually unstable. As a consequence, the user experience is negatively affected because the users are distracted during the tracking tasks.

As an alternative, it is possible to force a dynamic graph drawing to achieve visual stability. Multiple actors must be mapped to the same node position in the Euclidean Space at different points in time. Likewise, multiple relations must be mapped to the same edge position following the same principle. As a result, the dynamic graph drawing maintains a considerable number of elements on the screen during the temporal navigation, despite the fact that the network is actually changing. The visual stability of a dynamic graph drawing is a useful characteristic from the user's perspective. It improves the user experience during tracking tasks because the addition or removal of elements is minimal. Nevertheless, it is confusing to observe multiple actors appear in the same position at different points in time. This last has a negative impact on the efficiency of the visual search. The length of the scan paths was longer, while the number of fixations on target slightly decreased.

Introducing a restriction to prevent too many actors or relations from appearing in the same position, makes the dynamic graph drawing to achieve a moderate level of visual stability. Such a characteristic makes the changes more noticeable. However, they do not negatively affect the user experience during tracking tasks. In addition, the efficiency of the visual search is also improved. The number of fixations on target is considerably higher in comparison to those drawings with full visual stability. This suggests that there is a trade-off between the user experience and the efficiency of the visual search, which depends on the visual stability of the dynamic graph drawing.

Regarding visual stability and its applications in other types of metaphors, it is recommended to use features like the highlighting or scaling of elements to compensate the low accuracy of the visual search. A highlighted or scaled element can be tracked easier during the temporal navigation, while the visual stability of the metaphor keeps the distraction levels to the minimum.

Despite the achievements of this thesis, there are some aspects concerning the visual stability of a dynamic graph drawing that have not been addressed. For example, the mathematical model of visual stability was developed using the discrete representation of time. Therefore, it is not possible to quantify such a property in those dynamic graph drawings created with the continuous representation of time. Furthermore, the model of visual stability operates exclusively with dynamic graph drawings rather than any other type of metaphor. An extension to the model could lead to a more general approach to quantify the visual stability of any visual representation. Nonetheless, it would be necessary to conduct case studies in order to proceed in this direction. Lastly, the algorithms described in this thesis were designed to track actors or network

related attributes over time. An algorithm designed to track relation based patterns has never been explored in terms of the user experience or the efficiency of the visual search. The extended version of the model and the corresponding case studies combining eye-tracking devices as well as questionnaires could lead to specialized algorithms for efficiently tracking distinct types of patterns in dynamic networks.

Appendix A.

Questionnaire used in the evaluation of the circular layout, circular super graph, foresighted circular layout, flickering reduction and degree stabilization algorithms

Collaborative Learning in Intelligent Distributed Environments

University Of Duisburg-Essen

Rights of the participant

As a participant of this scientific experiment, you have the right to:

- Be treated with respect.
- Ask for additional information about the study.
- Leave at any moment.
- Request complete confidentiality about the provided information.
- Ask for the results of the study.
- Have a compensation for your help.

Collaborative Learning in Intelligent Distributed Environments

University Of Duisburg-Essen

Participation Agreement

The COLLIDE research group is investigating the effects of visual stability in the exploration of dynamic networks. The objective of the following study is to evaluate the users actions as he tries to track a set of actors over time.

It is our duty to inform that your participation will be completely anonymous. The participant will be asked to complete a series of tasks along with a small questionnaire at the end of the test. This experiment can be recorded in order to analyze the reactions of the user during each task. We guarantee that the recording will be used only for scientific purposes, respecting the confidentiality of the participant at all moment. It is worth to mention that the study takes around 30 minutes and if the participant does not feel comfortable the experiment can be stopped at any moment.

Having understood the terms, I accept to participate in this scientific experiment.

_____ _____

Participant Signature: Evaluator Signature:

Date:_____

Collaborative Learning in Intelligent Distributed Environments

University Of Duisburg-Essen

List of tasks : Visual Stability Study.

1. The following picture represents the first snapshot of the dynamic network. Please locate the following persons: *Tony Mountifield, Kevin P. Fleming and Rod Dorman.* Write down your observations once the task has been completed.

2. The next picture represents the seconds snapshot of the dynamic network. Do *Tony Mountifield, Kevin P. Fleming and Rod Dorman* appear in the current picture? Please write down your observations once the task has been completed.

3. Are the persons that appear in this state of the dynamic network located in the same layout position in comparison to the previous picture? Please write down your observations once the task has been completed.

4. Are the changes to the network structure distracting? Please write down your observations once the task has been completed.

5. The next picture represents the thirds snapshot of the dynamic network. Do *Tony Mountifield, Kevin P. Fleming and Rod Dorman* appear in the current picture? Please write down your observations once the task has been completed.

6. Are the persons that appear in this state of the dynamic network located in the same layout position in comparison to the previous picture? Please write down your observations once the task has been completed.

7. Are the changes to the network structure are distracting? Please write down your observations once the task has been completed.

8. The next picture represents the fourth snapshot of the dynamic network. Do *Tony Mountifield, Kevin P. Fleming and Rod Dorman* appear in the current picture? Please write down your observations once the task has been completed.

9. Are the persons that appear in this state of the dynamic network located in the same layout position in comparison to the previous picture? Please write down your observations once the task has been completed.

10. Are the changes to the network structure distracting? Please write down your observations once the task has been completed.

11. The next picture represents the last snapshot of the dynamic network. Do *Tony Mountifield, Kevin P. Fleming and Rod Dorman* appear in the current picture? Please write down your observations once the task has been completed.

12. Are the persons that appear in this state of the dynamic network located in the same layout position in comparison to the previous picture? Please write down your observations once the task has been completed.

13. Are the changes to the network structure distracting? Please write down your observations once the task has been completed.

Collaborative Learning in Intelligent Distributed Environments

University Of Duisburg-Essen

Questionnaire

A) Evaluate the previous tasks using the following scale:

Strongly Disagree				Strongly Agree
1	2	3	4	5

____ A person maintains its position over the layout as the network evolves.

____ I had to search for a specific person several times over the layout.

____ It is easy to locate a person with this layout.

____ The graph drawing technique is useful to track persons over time.

____ The network structure changes drastically.

____ As the network evolves, the number of nodes and edges appearing over the layout distract me.

B) What are the positive aspects of this network layout?

C) What are the negative aspects of this network layout?

Appendix B.

Questionnaire used in the evaluation of the foresighted heat ring

Collaborative Learning in Intelligent Distributed Environments

University Of Duisburg-Essen

Rights of the participant

As a participant of this scientific experiment, you have the right to:

- Be treated with respect.
- Ask for additional information about the study.
- Leave at any moment.
- Request complete confidentiality about the provided information.
- Ask for the results of the study.
- Have a compensation for your help.

Collaborative Learning in Intelligent Distributed Environments

University Of Duisburg-Essen

Participation Agreement

The COLLIDE research group is investigating the effects of visual stability in the exploration of dynamic networks. The objective of this experiment is to evaluate a recently developed technique to detect the changes on productive and collaborative researchers over time.

It is our duty to inform that your participation will be completely anonymous. The participant will be asked to complete a series of tasks along with a small questionnaire at the end of the test. This experiment can be recorded in order to analyze the reactions of the user during each task. We guarantee that the recording will be used only for scientific purposes, respecting the confidentiality of the participant at all moment. It is worth to mention that the study takes around 25 minutes and if the participant does not feel comfortable the experiment can be stopped at any moment.

Having understood the terms, I accept to participate in this scientific experiment.

_____ _____

Participant Signature: Evaluator Signature:

Date:_____

Collaborative Learning in Intelligent Distributed Environments

University Of Duisburg-Essen

Evaluation Tasks: Foresighted Heat Ring

1. On the top of the screen you will find the *menu bar*. Please list the different perspectives offered by this component and explore their features.

2. Located below the *menu bar,* you will find the *canvas.* Please describe the current drawing.

3. Move the mouse *over one of the labels surrounding* the foresighted heat ring. Please explain your observations.

4. Click on one of the labels surrounding the foresighted heat ring. Then, move the mouse away from it. Please write down your observations.

5. Click on the same label once again. Then, move the mouse away from it. What happens this time?

6. Move the mouse over one of the nodes inside the foresighted heat ring. Please write down your observations.

7. What can you say about the *small markers* inside the foresighted heat ring?

8. What can you say about the *bigger markers* inside the foresighted heat ring?

9. What does it mean to have a marker near the center of the foresighted heat ring?

10. What does it mean to have a marker far away from the center ?

11. Select the *time navigation* perspective from the menu bar. Then, using the *time frame selector* from the tool bar go to the *time frame 1*. Please write down your observations.

12. On the current drawing, select the researchers "GS" and "HW". Now navigate through the dynamic network starting from the *time frame 0* to *time frame 3*. What happens to the productivity and collaboration of the selected researchers?

13. What can you say about "GS" and "HW" ?

14. While you observed the evolution of "GS" and "HW", other researchers entered and exited the layout. Did you noticed these changes?

15. While you observed the evolution of "GS" and "HW", other researchers were changing their productivity and collaboration. Did the movement of their markers distract you from following "GS" and "HW"?

Collaborative Learning in Intelligent Distributed Environments

University Of Duisburg-Essen

Questionnaire

A) Evaluate the features offered by the visualization technique using the following scale:

Strongly Disagree				Strongly Agree
1	2	3	4	5

____The visualization is easy to understand.

____It is easy to select the labels surrounding the circular layout.

____It is easy to obtain information about the nodes inside the circular layout.

____The size of the nodes reflects the productivity of a researcher.

____The heat scale indicates the collaborative level of a researcher.

____The color of a selected node helps to detect faster the changes on the productivity and collaboration of researchers.

____The movement of the elements surrounding the circular layout are distracting.

____The movement of the non-selected elements inside the circular layout makes it difficult to see the evolution of specific researchers.

____The technique is useful to visualize the evolution of productive and collaborative researchers over time.

Bibliography

[1] Abello, J., Van Ham, F., and Krishnan, N. (2006). Ask-graphview: A large scale graph visualization system. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):669–676.

[2] Alba, R. D. (1973). A graph-theoretic definition of a sociometric clique. *Journal of Mathematical Sociology*, 3(1):113–126.

[3] Albert, W. and Tullis, T. (2010). *Measuring the user experience*. Morgan Kaufmann.

[4] Archambault, D. and Purchase, H. (2012). The mental map and memorability in dynamic graphs. In *Proceedings of the Pacific Visualization Symposium (PacificVis), 2012 IEEE*, pages 89–96.

[5] Archambault, D. and Purchase, H. (2013). Mental map preservation helps user orientation in dynamic graphs. In *Graph Drawing*, volume 7704 of *Lecture Notes in Computer Science*, pages 475–486. Springer Berlin Heidelberg.

[6] Auber, D. (2004). Tulip—a huge graph visualization framework. In *Graph Drawing Software*, pages 105–126. Springer.

[7] Bach, B., Pietriga, E., and Fekete, J.-D. (2014a). Graphdiaries: animated transitions and temporal navigation for dynamic networks. *IEEE Transactions on Visualization and Computer Graphics*, 20(5):740–754.

[8] Bach, B., Pietriga, E., Fekete, J.-D., et al. (2014b). Visualizing dynamic networks with matrix cubes. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*.

[9] Backstrom, L., Huttenlocher, D., Kleinberg, J., and Lan, X. (2006). Group formation in large social networks: membership, growth, and evolution. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 44–54. ACM.

[10] Bastian, M., Heymann, S., Jacomy, M., et al. (2009). Gephi: an open source software for exploring and manipulating networks. *ICWSM*, 8:361–362.

[11] Batini, C., Nardelli, E., and Tamassia, R. (1986). A layout algorithm for data flow diagrams. *IEEE Transactions on Software Engineering*, SE-12(4):538–546.

[12] Batini, C., Talamo, M., and Tamassia, R. (1984). Computer aided layout of entity relationship diagrams. *Journal of Systems and Software*, 4(2):163–173.

[13] Bavelas, A. (1950). Communication patterns in task-oriented groups. *Journal of the acoustical society of America*, 22:725–730.

[14] Bender-deMoll, S. and McFarland, D. A. (2006). The art and science of dynamic network visualization. *Journal of Social Structure*, 7(2):1–38.

[15] Blascheck, T., Kurzhals, K., Raschke, M., Burch, M., Weiskopf, D., and Ertl, T. (2014). State-of-the-art of visualization for eye tracking data. In *Proceedings of EuroVis*, volume 2014.

[16] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3:993–1022.

[17] Blignaut, P. (2010). Visual span and other parameters for the generation of heatmaps. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, pages 125–128. ACM.

[18] Böhringer, K.-F. and Paulisch, F. N. (1990). Using constraints to achieve stability in automatic graph layout algorithms. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, pages 43–51, New York, NY, USA. ACM.

[19] Borner, K., Glänzel, W., Scharnhorst, A., and den Besselaar, P. V. (2011). Modeling science: studying the structure and dynamics of science. *Scientometrics*, 89(1):347–348.

[20] Borner, K.rner, K., Huang, W., Linnemeier, M., Duhon, R., Phillips, P., Ma, N., Zoss, A., Guo, H., and Price, M. (2010). Rete-netzwerk-red: Analyzing and visualizing scholarly networks using the network workbench tool. *Scientometrics*, 83(3):863–876.

[21] Boyack, K. W., Börner, K., and Klavans, R. (2009). Mapping the structure and evolution of chemistry research. *Scientometrics*, 79(1):45–60.

[22] Brandes, U., Indlekofer, N., and Mader, M. (2012). Visualization methods for longitudinal social networks and stochastic actor-oriented modeling. *Social Networks*, 34(3):291–308.

[23] Bridgeman, B., Hendry, D., and Stark, L. (1975). Failure to detect displacement of the visual world during saccadic eye movements. *Vision research*, 15(6):719–722.

[24] Bridgeman, B. and Stark, L. (1979). Omnidirectional increase in threshold for image shifts during saccadic eye movements. *Attention, Perception, and Psychophysics*, 25(3):241–243.

[25] Bridgeman, B., Van der Heijden, A., and Velichkovsky, B. M. (1994). A theory of visual stability across saccadic eye movements. *Behavioral and Brain Sciences*, 17(2):247–257.

[26] Byrne, M. D., Anderson, J. R., Douglass, S., and Matessa, M. (1999). Eye tracking the visual search of click-down menus. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 402–409. ACM.

[27] Chan, E. P. F. and Lochovsky, F. H. (1980). A graphical database design aid using the entity-relationship model. In *Proceedings of the 1st International Conference on the Entity-Relationship Approach to Systems Analysis and Design*, pages 295–310, Amsterdam, The Netherlands, The Netherlands. North-Holland Publishing Co.

[28] Correa, C. and Ma, K.-L. (2011). Visualizing social networks. In Aggarwal, C. C., editor, *Social Network Data Analytics*, pages 307–326. Springer US.

[29] Crucitti, P., Latora, V., and Porta, S. (2006). Centrality in networks of urban streets. *Chaos: an interdisciplinary journal of nonlinear science*, 16(1):015113.

[30] Deubel, H., Bridgeman, B., and Schneider, W. X. (2004). Different effects of eyelid blinks and target blanking on saccadic suppression of displacement. *Perception and Psychophysics*, 66(5):772–778.

[31] Diehl, S., Görg, C., and Kerren, A. (2001). Preserving the mental map using foresighted layout. In Ebert, D., Favre, J., and Peikert, R., editors, *Data Visualization 2001*, Eurographics, pages 175–184. Springer Vienna.

[32] Diehl, S. and Görg, C. (2002). Graphs, they are changing - dynamic graph drawing for a sequence of graphs. In *Proceedings of the 10th Int. Symp. Graph Drawing (GD 2002), number 2528 in Lecture Notes in Computer Science, LNCS*, pages 23–31. Springer-Verlag.

[33] Dill, J., Earnshaw, R., Kasik, D., Vince, J., and Wong, P. (2012). *Expanding the Frontiers of Visual Analytics and Visualization*. SpringerLink : Bücher. Springer London.

[34] Dorogovtsev, S. N., Goltsev, A. V., and Mendes, J. F. F. (2006). K-core organization of complex networks. *Physical review letters*, 96(4):040601.

[35] Dunne, C. and Shneiderman, B. (2009). Improving graph drawing readability by incorporating readability metrics: A software tool for network analysts. Technical report, University of Maryland.

[36] Dwyer, T., Koren, Y., and Marriott, K. (2009). Constrained graph layout by stress majorization and gradient projection. *Discrete Mathematics*, 309(7):1895–1908.

[37] Eades, P. (1984). A heuristics for graph drawing. *Congressus numerantium*, 42:146–160.

[38] Eades, P., Lai, W., Misue, K., and Sugiyama, K. (1991). *Preserving the mental map of a diagram*. International Institute for Advanced Study of Social Information Science, Fujitsu Limited.

[39] Eisenberg, J. D. and Bellamy-Royds, A. (2014). *SVG essentials*. " O'Reilly Media, Inc.".

[40] Erten, C., Harding, P. J., Kobourov, S. G., Wampler, K., and Yee, G. (2004). Graphael: Graph animations with evolving layouts. In *Proceedings of Graph Drawing*, pages 98–110. Springer.

[41] Fafchamps, M. and van der Leij, M. J. (2006). Scientific networks and co-authorship. Economics Series Working Papers 256, University of Oxford, Department of Economics.

[42] Festiger, L. (1949). The analysis of sociograms using matrix algebra. *Human relations*.

[43] Field, A. (2009). *Discovering Statistics Using SPSS*. ISM (London, England). SAGE Publications.

[44] Freeman, L. C. (1978). Centrality in social networks conceptual clarification. *Social Networks*, page 215.

[45] Freeman, L. C., Borgatti, S. P., and White, D. R. (1991). Centrality in valued graphs: A measure of betweenness based on network flow. *Social networks*, 13(2):141–154.

[46] Fruchterman, T. M. J. and Reingold, E. M. (1991). Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164.

[47] Gansner, E. R., Hu, Y., and North, S. (2013). Visualizing streaming text data with dynamic graphs and maps. In *Proceedings of the 20th International Conference on Graph Drawing*, GD'12, pages 439–450, Berlin, Heidelberg. Springer-Verlag.

[48] Gansner, E. R. and Koren, Y. (2007). Improved circular layouts. In *Proceedings of Graph Drawing*, pages 386–398. Springer.

[49] Gansner, E. R., North, S. C., and Vo, K.-P. (1988). Dag—a program that draws directed graphs. *Software: Practice and Experience*, 18(11):1047–1062.

[50] Gelernter, D. (1985). Generative communication in linda. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 7(1):80–112.

[51] Ghani, S., Elmqvist, N., and Yi, J. S. (2012). Perception of animated node-link diagrams for dynamic graphs. In *Proceedings of the Computer Graphics Forum*, volume 31, pages 1205–1214. Wiley Online Library.

[52] Gloor, P. and Zhao, Y. (2004). A temporal communication flow visualizer for social network analysis. In *Proceedings of the Computer Supported Collaborative Work Conference CSCW'04, Workshop on Social Networks*, Chicago, U.S.A.

[53] Gloor, P. A., Laubacher, R., Zhao, Y., and Dynes, S. B. C. (2004). Temporal visualization and analysis of social networks. In *Proceedings of the NAACSOS Conference, June 27 - 29, Pittsburgh PA, North American Association for Computational Social and Organizational Science*. In.

[54] Göhnert, T., Harrer, A., Hecking, T., and Hoppe, H. (2014). A workbench for visual design of executable and re-usable network analysis workflows. In Kawash, J., editor, *Online Social Media Analysis and Visualization*, Lecture Notes in Social Networks, pages 181–199. Springer International Publishing.

[55] Göhnert, T., Harrer, A., Hecking, T., and Hoppe, H. U. (2013). A workbench to construct and re-use network analysis workflows - concept, implementation, and example case. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, Niagara Falls, Canada.

[56] Goldberg, J. H. and Kotval, X. P. (1999). Computer interface evaluation using eye movements: methods and constructs. *International Journal of Industrial Ergonomics*, 24(6):631–645.

[57] Goldberg, J. H., Stimson, M. J., Lewenstein, M., Scott, N., and Wichansky, A. M. (2002). Eye tracking in web search tasks: design implications. In *Proceedings of the 2002 symposium on Eye tracking research & applications*, pages 51–58. ACM.

[58] Hansen, D., Shneiderman, B., and Smith, M. A. (2010). *Analyzing social media networks with NodeXL: Insights from a connected world.* Morgan Kaufmann.

[59] Harrer, A., Zeini, S., and Ziebarth, S. (2009). Integrated representation and visualisation of the dynamics in computer-mediated social networks. In *Proceedings of the 2009 International Conference on Advances in Social Network Analysis and Mining*, ASONAM '09, pages 261–266, Washington, DC, USA. IEEE Computer Society.

[60] Hecking, T., Göhnert, T., Zeini, S., and Hoppe, U. (2013). Task and time aware community detection in dynamically evolving social networks. *Procedia Computer Science*, 18:2066–2075.

[61] Hecking, T., Ziebarth, S., and Hoppe, H. U. (2014). Analysis of dynamic resource access patterns in a blended learning course. In *Proceedings of the Fourth International Conference on Learning Analytics and Knowledge*, pages 173–182. ACM.

[62] Horak, Z., Kudelka, M., Snasel, V., Abraham, A., and Rezankova, H. (2011). Forcoa.net: An interactive tool for exploring the significance of authorship networks in dblp data. In *Proceedings of the International Conference on Computational Aspects of Social Networks (CASoN)*, pages 261–266.

[63] Huang, W. (2007). Using eye tracking to investigate graph layout effects. In *Proceedings of the International Asia-Pacific Symposium on Visualization*, pages 97–100. IEEE.

[64] Jacob, R. and Karn, K. S. (2003). Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. *Mind*, 2(3):4.

[65] Jacomy, M., Heymann, S., Venturini, T., and Bastian, M. (2011). Forceatlas2, a continuous graph layout algorithm for handy network visualization. *Medialab center of research*, 560.

[66] Jakobsen, T. (2001). Advanced character physics. In *Proceedings of the Game Developers Conference*, pages 383–401.

[67] Kamada, T. and Kawai, S. (1989). An algorithm for drawing general undirected graphs. *Information processing letters*, 31(1):7–15.

[68] Kobourov, S. G. (2004). Force-directed drawing algorithms. In *Handbook of Graph Drawing and Visualization*, chapter 12, pages 383–408. CRC Press.

[69] Krempel, L. (2005). *Visualisierung komplexer Strukturen - Grundlagen der Darstellung mehrdimensionaler Netzwerke.* Campus-Verl.

[70] Kudelka, M., Horak, Z., Snasel, V., and Abraham, A. (2010). Social network reduction based on stability. In *Proceedings of the International Conference on Computational Aspects of Social Networks (CASoN)*, pages 509–514.

[71] Kumpula, J. M., Kivelä, M., Kaski, K., and Saramäki, J. (2008). Sequential algorithm for fast clique percolation. *Physical Review E*, 78(2):026109.

[72] Latora, V. and Marchiori, M. (2001). Efficient behavior of small-world networks. *Physical review letters*, 87(19):198701.

[73] Latora, V. and Marchiori, M. (2004). A measure of centrality based on the network efficiency. arxiv: con-math 0402050.

[74] Lawrence, S., Robert, K., Susan, S., Derek, H., and Bruce, B. (1976). Saccadic suppression of image displacement. *Vision Research*, 16(10):1185–1187.

[75] Lee, S. and Bozeman, B. (2005). The impact of research collaboration on scientific productivity. *Social Studies of Science*, 35(5):673–702.

[76] Lehmann, S., Schwartz, M., and Hansen, L. K. (2008). Biclique communities. *Physical Review E*, 78(1):016108.

[77] Likert, R. (1932). A technique for the measurement of attitudes. *Archives of psychology*.

[78] Liversedge, S., Gilchrist, I., and Everling, S. (2011). *The Oxford handbook of eye movements*. Oxford University Press.

[79] Loeliger, E., Nehaniv, C. L., and Munro, A. J. (2012). Heat-maps and visualization for heterogeneous biomedical data based on information distance geometry. In *Proceedings of the 9^{th} international conference on Information Processing in Cells and Tissues*, IPCAT'12, pages 183–187, Berlin, Heidelberg. Springer-Verlag.

[80] Luce, R. D. and Perry, A. D. (1949). A method of matrix analysis of group structure. *Psychometrika*, 14(2):95–116.

[81] Mack, A. (1970). An investigation of the relationship between eye and retinal image movement in the perception of movement. *Perception and Psychophysics*, 8(5):291–298.

[82] MacKay, D. M. (1973). Visual stability and voluntary eye movements. In *Central Processing of Visual Information A: Integrative Functions and Comparative Data*, pages 307–331. Springer.

[83] Mcconkie, G. W. and Currie, C. B. (1996). Visual stability across saccades while viewing complex pictures. *Journal of Experimental Psychology: Human Perception and Performance*, 22(3):563–581.

[84] Mello-Thoms, C., Nodine, C. F., and Kundel, H. L. (2002). What attracts the eye to the location of missed and reported breast cancers? In *Proceedings of the 2002 symposium on Eye tracking research & applications*, pages 111–117. ACM.

[85] Messinger, E. B. (1988). *Automatic Layout of Large Directed Graphs*. PhD thesis, University of Washington. AAI8906930.

[86] Misue, K., Eades, P., Lai, W., and Sugiyama, K. (1995). Layout adjustment and the mental map. *Journal of Visual Languages & Computing*, 6(2):183–210.

[87] Mokken, R. J. (1979). Cliques, clubs and clans. *Quality and Quantity*, 13(2):161–173.

[88] Moody, James, D. A. M. and Bender-DeMoll, S. (2005). Dynamic network visualization: Methods for meaning with longitudinal network movies. *American Journal of Sociology*.

[89] Moreno, J. (1934). *Who shall survive? : a new approach to the problem of Human Interrelations*, volume 58 of *Nervous and mental disease monograph series*. Nervous and Mental Disease Publ., Washington.

[90] O'Regan, J. K. (1992). Solving the" real" mysteries of visual perception: the world as an outside memory. *Canadian Journal of Psychology/Revue canadienne de psychologie*, 46(3):461.

[91] Palla, G., Barabási, A.-L., and Vicsek, T. (2007). Quantifying social group evolution. *Nature*, 446(7136):664–667.

[92] Palla, G., Derényi, I., Farkas, I., and Vicsek, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818.

[93] Pohl, M., Schmitt, M., and Diehl, S. (2009). Comparing the readability of graph layouts using eye-tracking and task-oriented analysis. In *Proceedings of the Fifth Eurographics conference on Computational Aesthetics in Graphics, Visualization and Imaging*, pages 49–56. Eurographics Association.

[94] Poole, A. and Ball, L. J. (2005). Eye tracking in human-computer interaction and usability research: Current status and future. In *Prospects", Chapter in C. Ghaoui (Ed.): Encyclopedia of Human-Computer Interaction. Pennsylvania: Idea Group, Inc.*

[95] Poole, A. and Ball, L. J. (2006). Eye tracking in hci and usability research. *Encyclopedia of human-computer interaction*, 1:211–219.

[96] Poole, A., Ball, L. J., and Phillips, P. (2005). In search of salience: A response-time and eye-movement analysis of bookmark recognition. In *People and Computers XVIII—Design for Life*, pages 363–378. Springer.

[97] Pretorius, M. C., Calitz, A. P., and van Greunen, D. (2005). The added value of eye tracking in the usability evaluation of a network management tool. In *Proceedings of the 2005 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*, pages 1–10. South African Institute for Computer Scientists and Information Technologists.

[98] Purchase, H. and Samra, A. (2008). Extremes are better: Investigating mental map preservation in dynamic graphs. In Stapleton, G., Howse, J., and Lee, J., editors, *Diagrammatic Representation and Inference*, volume 5223 of *Lecture Notes in Computer Science*, pages 60–73. Springer Berlin Heidelberg.

[99] Purchase, H. C., Hoggan, E., and Görg, C. (2007). How important is the "mental map"?–an empirical investigation of a dynamic graph layout algorithm. In *Proceedings of Graph drawing*, pages 184–195. Springer.

[100] Ramos, A., Chounta, I.-A., Göhnert, T., and Hoppe, U. (2015a). Exploring visual stability in dynamic graph drawings: A case study. In *Proceedings of the 2015 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM)*.

[101] Ramos, A., Chounta, I.-A., Göhnert, T., and Hoppe, U. (2015b). Using visual stability to support search efficiency and user experience in dynamic graph drawings. In *Proceedings of the 2015 Second European Network Intelligence Conference, ENIC 2015, Karlskrona, Sweden, September 21-22, 2015*, pages 145–149.

[102] Ramos, A., Chounta, I.-A., Göhnert, T., and Hoppe, U. (2015c). Visual stability in dynamic graph drawings. *Journal of Interactive Media (I-COM)*.

[103] Ramos, A., Gohnert, T., Zeini, S., and Hoppe, U. (2014). Foresighted heat ring: Tracking the productivity and collaborativeness of researchers in dynamic networks. In *Proceedings of the 2014 European Network Intelligence Conference, ENIC 2014, Wroclaw, Poland, September 29-30, 2014*, pages 100–105.

[104] Ramos, A., Sánchez, J. A., and Hernández-Bolaños, F. (2010). Ontostarfish: visualization of collaboration networks using starfields, ontologies and fisheye views. In *Proceedings of the 3rd Mexican Workshop on Human-Computer Interaction*, MexIHC '10, pages 44–53.

[105] Robins, G. (1988). The isi grapher: A portable tool for displaying graphs pictorially. *Multicomputer Vision*, page 185.

[106] Robinson, A. C. and Center, G. (2006). Highlighting techniques to support geovisualization. In *Proceedings of the ICA Workshop on Geovisualization and Visual Analytics*.

[107] Rowe, L. A., Davis, M., Messinger, E., Meyer, C., Spirakis, C., and Tuan, A. (1987). A browser for directed graphs. *Software: Practice and Experience*, 17(1):61–76.

[108] Seidman, S. B. (1983). Network structure and minimum degree. *Social networks*, 5(3):269–287.

[109] Seidman, S. B. and Foster, B. L. (1978). A graph-theoretic generalization of the clique concept. *Journal of Mathematical sociology*, 6(1):139–154.

[110] Spearman, C. (1904). The proof and measurement of association between two things. *The American journal of psychology*, 15(1):72–101.

[111] Tamassia, R. (2007). *Handbook of Graph Drawing and Visualization*. CRC Press.

[112] Tamassia, R., Battista, G. D., and Batini, C. (1988). Automatic graph drawing and readability of diagrams. *IEEE Transactions on Systems, Man and Cybernetics*, 18(1):61–79.

[113] Thomas, J. J. and Cook, K. A. (2005). Illuminating the path: The research and development agenda for visual analytics. Technical report, Pacific Northwest National Laboratory (PNNL), Richland, WA (US).

[114] Trickey, H. (1988). Drag: A graph drawing system. In *Proceedings of the International Conference on Electronic Publishing, Document Manipulation and Typography*, pages 171–182.

[115] Trier, M. and Bobrik, A. (2007). Analyzing the dynamics of community formation using brokering activities. In Steinfield, C., Pentland, B., Ackerman, M., and Contractor, N., editors, *Communities and Technologies 2007*, pages 463–477. Springer London.

[116] von Holst, E. and Mittelstaedt, H. (1950). Das reafferenzprinzip. wechselwirkungen zwischen zentralnervensystem und peripherie. *Naturwissenschaften*, 37:464–476.

[117] Wasserman, S. and Faust, K. (1994). *Social network analysis: Methods and applications*, volume 8. Cambridge university press.

[118] Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442.

[119] Weinbrenner, S. (2012). *SQLSpaces – A Platform for Flexible Language-Heterogeneous Multi-Agent Systems*. PhD thesis, Universität Duisburg-Essen.

[120] Whipple, W. R. and Wallach, H. (1978). Direction-specific motion thresholds for abnormal image shifts during saccadic eye movement. *Perception and psychophysics*, 24(4):349–355.

[121] Wilkinson, L. and Friendly, M. (2009). The history of the cluster heat map. *The American Statistician*, 63(2).

[122] Wilson, A. G. (2000). *Complex spatial systems: the modeling foundations of urban and regional analysis*. Pearson Education.

[123] Wong, P. C. and Thomas, J. (2004). Visual analytics. *IEEE Computer Graphics and Applications*, 24(5):20–21.

[124] Yee, K.-P., Fisher, D., Dhamija, R., and Hearst, M. (2001). Animated exploration of dynamic graphs with a radial layout. In *Proceedings of the IEEE Symposium on Information Visualization*.

[125] Zahra, S. B. and Hussain, T. (2014). An approach of a gradient in graphics: Css3. *European Journal of Computer Science and Information Technology*, 2(3):10–16.

[126] Zeini, S., Göhnert, T., Hecking, T., Krempel, L., and Hoppe, H. U. (2014). The impact of measurement time on subgroup detection in online communities. In *State of the Art Applications of Social Network Analysis*, pages 249–268. Springer.